



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

---

---

**FACULTAD DE INGENIERIA**

**“Desarrollo de un Sistema Administrador de  
Información para una Institución Educativa”**

**T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A N :  
JUAN EDUARDO CASTAÑEDA VAZQUEZ  
MIGUEL ANGEL HERNÁNDEZ GARCIA  
LUIS ANTONIO VAZQUEZ RAMOS**



**MEXICO D.F.**

**DIRECTOR DE TESIS M. EN I. LAURO SANTIAGO CRUZ**

**2003**

# Índice General

## Índice

### Capítulo 1 - Generalidades

1.1. Conceptos fundamentales de bases de datos.....	1
1.1.1. Base de datos relacional.....	3
1.1.2. Lenguaje de consulta estructurado.....	4
1.1.3. Integridad referencial y normalización.....	4
1.2. Programación Orientada a objetos.....	6
1.2.1. Conceptos fundamentales de la POO.....	6
1.2.2. Características de la POO.....	7
1.2.3. Ventajas de los lenguajes orientados a objetos.....	8
1.3. Plataforma de ambiente y desarrollo.....	9
1.3.1. .NET Framework.....	9
1.3.2. Visual Basic .NET.....	12
1.3.3. Arquitecturas de aplicación.....	13
1.3.4. SQL Server.....	15

### Capítulo 2 Análisis del Sistema

2.1. Situación actual.....	18
2.2. Definición de la problemática.....	51
2.3. Propuesta de solución y definición del alcance del proyecto..	54

### Capítulo 3 - Diseño del Sistema

3.1. Modelado del sistema.....	56
3.2. Estructura de datos.....	61

## **Capítulo 4 - Generador de Código**

4.1. Definición del generador de código.....	106
4.1.1. Características del generador de código.....	106
4.2. Funcionalidad.....	108
4.2.1. Módulo nuevo / abrir.....	111
4.2.2. Módulo de conexión a datos.....	121
4.2.3. Módulo de configuración.....	123
4.2.4. Módulo de generación de código .....	134
4.3. Generación del código base .....	134
4.3.1. Interfaz del usuario .....	135
4.3.2. Componentes de acceso a datos .....	142

## **Capítulo 5 Desarrollo e Implementación**

5.1. Construcción del sistema de información.....	164
5.1.1. Muestra de la configuración del sistema.....	166
5.2. Descripción grafica del sistema .....	174
5.3. Pruebas e instalación del sistema .....	192

<b>Resultados y Conclusiones.....</b>	<b>203</b>
---------------------------------------	------------

<b>Bibliografía.....</b>	<b>208</b>
--------------------------	------------

## Introducción

---

En la actualidad el uso de las tecnologías informáticas en la administración y gestión de los procesos en todos los ámbitos se ha vuelto muy común, podemos encontrar actualmente sistemas de gestión tan sencillos, como los usados para calcular el inventario en un almacén o llevar la contabilidad en una empresa, hasta los sistemas que controlan el tráfico aéreo o los que son usados en campos como la ciencia.

La demanda de soluciones a los problemas presentados por estos y otros ámbitos en tiempos razonablemente cortos, haciendo uso de las tecnologías informáticas, ha traído como consecuencia la necesidad de reducir los tiempos de desarrollo de dichas soluciones; se han inventado diversos modelos siempre con la finalidad de entregar al cliente un producto de excelente calidad y en un tiempo menor, esto con el objetivo principal de disminuir el costo de los proyectos.

Actualmente el uso de Herramientas RAD (*Rapid Application Deployment*) y CASE (*Computer Aided Software Engineering*) han ayudado a disminuir los tiempos de desarrollo de los sistemas, buscando enfocarse más en las etapas de Análisis y Diseño de la aplicación, así como la organización y construcción de la Base De Datos usada por el sistema. Sin embargo, son pocas las soluciones existentes en el mercado mediante las cuales podemos disminuir los tiempos de desarrollo de forma drástica, consiguiendo automatizar los procesos de codificación de los sistemas.



Existen varias empresas e instituciones educativas cuyos procesos administrativos son anticuados y obsoletos, lo cual implica un deficiente manejo de sus recursos. El trabajo que a continuación se presenta surge como una necesidad de una Universidad, en la cual procesos como la administración de la información es deficiente, debido a que todos los trámites se realizan en papel, provocando que algunos documentos sean difíciles de encontrar y en algunos casos se pierdan.

Esta Universidad requiere contar con un sistema de administración de información de fácil manejo que satisfaga sus necesidades en cuanto a la administración de sus recursos. Por lo que, el objetivo central es desarrollar dicho sistema para solventar los problemas de almacenamiento y búsqueda de información que presentan en la actualidad, basándonos en la creación de una herramienta de generación de código, la cual disminuirá sustancialmente el tiempo de desarrollo.

El desarrollo de este sistema tiene la importancia de que solventa una necesidad real y la ventaja de que el código será generado por una herramienta que desarrollaremos, que tiene como característica principal la no exclusividad de la estructura definida en cualquier base de datos relacional, por lo cual puede ser utilizada para la creación de otros sistemas.

En la actualidad, para la construcción de sistemas, la etapa de desarrollo resulta una fase necesaria que siempre establece un tiempo proporcional de acuerdo al tipo y al alcance del código. En este sentido, con la construcción de la herramienta generadora de código, el tiempo de la fase de desarrollo del sistema administrador de información se verá drásticamente disminuido, comparado con el tiempo promedio calculado en el desarrollo sin el uso de la herramienta.

Una vez dejando claro el objetivo central de este trabajo y la justificación del mismo, es necesario mencionar el tipo de metodología que aplicaremos para ello.

Tanto para la creación del sistema administrador de información como para el “Generador de código” se utilizara la metodología clásica de desarrollo de sistemas, la cual comprende las etapas de análisis, diseño, desarrollo, pruebas e implementación.

Este trabajo se organiza en siete capítulos que podemos resumir de la siguiente forma.

## **Narrativa por Capítulo**

### *Capítulo 1 – Generalidades*

En este capítulo se presenta un marco teórico de los fundamentos de las bases de datos relacionales, la definición de éstas, su integridad y su normalización. Además se proporciona una introducción a la programación orientada a objetos, conceptos, principios y características que forman el fundamento para la comprensión de esta tecnología. Se presenta un panorama general de la plataforma Microsoft Framework .NET, mediante la cual se desarrollará el sistema de administración de información,

basados principalmente en Visual Basic .NET como lenguaje de programación. Se expondrán las tendencias y evolución de los modelos de solución existentes hoy en día para el desarrollo de sistemas, así como las ventajas y desventajas que generan en el desarrollo, implementación y mantenimiento de los mismos y finalmente hablaremos brevemente del sistema manejador de base de datos SQL Server que se utilizara para el almacenamiento de la información.

### *Capítulo 2 – Análisis del Sistema*

En este capítulo hacemos una revisión de la situación actual en la que se encuentra la Institución Educativa, mostrando un panorama general de sus principales actividades. Posteriormente definimos la problemática que presenta la institución en la administración de la información, basándonos en los procedimientos que actualmente utilizan para llevar a cabo sus actividades. Finalmente, planteamos la propuesta de solución y los alcances del proyecto.

### *Capítulo 3 – Diseño del Sistema.*

La etapa de diseño comprende la definición del modelado del sistema, así como la estructura de los datos y la forma en que éstos se relacionan. A partir de la descripción de las funciones y de la especificación de las interfaces es posible implementar cada parte que compone el sistema. En este capítulo, de acuerdo a los requisitos del sistema, crearemos la estructura de datos y la representación de el sistema; especificando la descripción de los datos y definiendo el comportamiento de la aplicación.

### *Capítulo 4 – Generador de Código*

Las herramientas de programación permiten a los desarrolladores de software editar, interpretar, compilar, probar y depurar, todos los programas fuentes de una aplicación o un sistema además de ayudar a la detección y comparación de errores, todo con la finalidad de que cada vez sea más fácil, rápido y agradable el desarrollo de los programas. En este capítulo se presenta la definición de la funcionalidad, el modelado de datos y objetos, las estructuras de datos y el desarrollo e implementación de la herramienta de generación de código que se usara para el desarrollo del Sistema Administrador de Información.

### *Capítulo 5 – Desarrollo e Implementación*

El capítulo Desarrollo e Implementación trata sobre la construcción del sistema e implementación de acuerdo al planteamiento descrito en el diseño del sistema, contempla la construcción del sistema utilizando el generador de código, la implementación en un ambiente real y la realización de pruebas.

Finalmente se presentan los resultados y conclusiones del trabajo completo y la bibliografía consultada.



---

## Generalidades

---

En este capítulo se presenta un marco teórico de los fundamentos de las bases de datos relacionales, la definición de éstas, su integridad y su normalización. Además se proporciona una introducción a la programación orientada a objetos, conceptos, principios y características que forman el fundamento para la comprensión de esta tecnología. Se presenta un panorama general de la plataforma Microsoft Framework .NET, mediante la cual se desarrollara el sistema de administración de información, basados principalmente en Visual Basic .NET como lenguaje de programación. Se expondrán las tendencias y evolución de los modelos de solución existentes hoy en día para el desarrollo de sistemas, así como las ventajas y desventajas que generan en el desarrollo, implementación y mantenimiento de los mismos y finalmente hablaremos brevemente del sistema manejador de base de datos SQL Server que se utilizara para el almacenamiento de la información.

### 1.1. Conceptos fundamentales de Base de Datos

Una base de datos es una colección de datos relacionados, organizados de cierta manera predefinida y con algunos programas que administren esa información y

---

su procesamiento<sup>1</sup>. Estos programas se conocen como el DBMS (*Data Base Management System*) y son los encargados de supervisar que los datos se encuentren almacenados correctamente, en forma segura y de brindar las herramientas para poder acceder a los datos.

Una Base de Datos debe garantizar que podrá almacenar los datos que recibe, pero que lo hará en forma eficiente y segura. Es decir, que su DBMS buscará la mejor forma de mantener los datos y que garantizará que la información se mantenga consistente y que no se altere a pesar de fallos que puedan ocurrir (daños en discos, fallo del suministro eléctrico, etc.).

El objetivo principal de un DBMS es facilitar el acceso a los datos, brindando herramientas ágiles, sencillas y realizando las tareas de recuperación y modificación de los datos. Es un conjunto de programas que tiene a su cargo tareas como:

Interacción con el sistema de archivos del sistema operativo

- Implementación de la integridad
- Implementación de la seguridad
- Respaldo y recuperación
- Control de concurrencia

Con el avance de la tecnología de cómputo, la necesidad de manejar información en base de datos aumentó, para poder hacerlo se crearon “modelo de datos” para describir los datos en los niveles conceptual y de visión. Principalmente podemos hablar de tres modelos, el de red, el jerárquico y el relacional. A continuación mostramos una breve descripción.

#### *Modelo de red*

Los datos en el modelo de red se representan por medio de registros y las relaciones entre los datos se representan con ligas, que pueden considerarse apuntadores. Los registros de la base de datos se organizan en forma de conjuntos de gráficas arbitrarias.

#### *Modelo jerárquico*

El modelo jerárquico es similar al modelo de red, en cuanto a que los datos y las relaciones entre los datos se representan por medio de registros y ligas, respectivamente. El modelo jerárquico difiere del de red en que los registros están organizados como conjuntos de árboles en vez de gráficas arbitrarias.

---

<sup>1</sup> Curtis L. Smith y Michael C. Amundsen, Programación de Base de Datos con Visual Basic, Prentice Hall, México, 1999.

### *Modelo relacional*

En el modelo relacional los datos y las relaciones entre los datos se representan por medio de una serie de tablas, las cuales tienen varias columnas con nombres únicos. Éste modelo es el que será utilizado en el desarrollo de la base de datos para el sistema de información.

#### **1.1.1. Base de Datos Relacional**

Las bases de datos relacionales son un tipo de Base de Datos que se definen como una colección de datos cuyos elementos tienen vínculos entre sí, compone su estructura principalmente en dos elementos, entidades y relaciones.

#### **Entidades**

Las entidades contienen atributos que pueden ser obligatorios u opcionales, debe identificarse por un nombre en singular. Se convertirán en las tablas de la Base de Datos, estas tablas se componen de arreglos de columnas y filas.

Existen dos clases de entidades, las entidades fuertes y débiles; las entidades fuertes dependen de sí mismas, mientras que las débiles dependen de la existencia de al menos de una entidad fuerte de la cual depende.

Las entidades poseen propiedades o características llamadas atributos. Entre todos los atributos de un tipo de entidad se debe elegir a uno, o varios, que identifiquen única y por lo menos cada una de las ocurrencias de este tipo de entidad.

#### **Relaciones**

Las relaciones son asociaciones o correspondencias entre entidades. Toda relación cuenta con tres características: nombre, que es el que le identifica de forma única y mediante el cual se referencia; grado, que identifica el número de tipos de entidad que participan en un tipo de relación y tipo de correspondencia, que especifica el tipo de relación entre las entidades y que puede tomar una de las siguientes formas:

- Uno a uno. Cuando en la relación sólo puede aparecer como máximo una ocurrencia del tipo de entidad por cada ocurrencia del otro tipo de entidad que interviene en la relación.
- Uno a muchos. Cuando para uno de los tipos de entidad puede haber un número indefinido de ocurrencias del otro tipo de entidad que interviene en la relación.

- Muchos a muchos. Cuando para uno de los tipos de entidad puede haber un número indefinido de ocurrencias del otro tipo de entidad.

### 1.1.2. Lenguaje de Consulta Estructurado

El Lenguaje de Consulta Estructurado, SQL (*Structure Query Language*), se basa en instrucciones para el manejo de la información de base de datos. Las instrucciones SQL se encuentran organizadas en dos grandes categorías:

- Instrucciones para la manipulación de datos (DML, *Data Model Language*). Estas instrucciones se usan principalmente para acceder a la información, estableciendo tareas como consultas, agregación, modificación y eliminación de los datos.
- Instrucciones para la definición de datos (DDL, *Data Definition Language*). Estas instrucciones se usan para crear, modificar y eliminar las estructuras de los distintos objetos de la base de datos.

### 1.1.3. Integridad Referencial y Normalización

El diseño de una base de datos, mediante el modelo relacional, comprende utilizar el conjunto de reglas de Integridad Referencial y aplicar la teoría de Normalización para obtener como resultado una base de datos confiable.

#### Integridad Referencial

Como parte de la seguridad para garantizar que la información sea consistente y correcta, así como para facilitar las relaciones entre la información de la Base de Datos, se implementan algunas restricciones de integridad. Estas son reglas que el DBMS valida y que obliga a cumplir. El diseñador de la Base de Datos decide cuáles reglas se programarán.

Existen varios tipos de restricciones de integridad, tales como lo son las llaves y la cardinalidad; las llaves (primarias, foráneas y únicas) evitan que se almacenen filas inconsistentes, mientras que la cardinalidad restringe el conjunto de relaciones permitidas entre entidades (uno a muchos, muchos a muchos, etc.).

Otros tipos de restricciones son:

- De dominio: Limita la cantidad de valores posibles para una columna.

- De valores nulos: Un valor nulo es un valor vacío o la ausencia de valor alguno en la columna. Cuando se dice que una columna es *NOT NULL*, quiere decir que la fila siempre tendrá un valor válido en esa columna. Es distinto de un blanco en una columna de tipo cadena o de un cero en una numérica. Este tiene sus propias características y una importante es que cuando se compara cualquier valor contra una columna nula, la comparación siempre será falsa.
- De afirmaciones: Son reglas específicas que se implementan en el ámbito de columnas. Por ejemplo, podríamos decir que una columna de número de alumnos en un curso no puede valer menos que cero o que el saldo en una cuenta nunca puede ser menor a 100 mil.

## Normalización

La normalización de datos, o normalización como tal, es un proceso de refinamiento de las estructuras de la base de datos para mejorar la velocidad a la que los datos puedan accederse, así como mejorar su integridad. En una base de datos bien normalizada se debe encontrar el punto medio entre velocidad e integridad<sup>2</sup>.

La normalización se lleva a cabo en una serie de cinco pasos. Cada paso corresponde a una forma normal que tiene propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto y, por lo tanto, son menos vulnerables a las anomalías de las actualizaciones. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal, las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal de las cinco propuestas.

Las primeras tres formas tienen que ver con la eliminación de datos repetitivos, redundantes o sin relación entre sí. Las últimas dos formas se centran en aislar varias relaciones para mejorar la integridad en conjunto de la base de datos.

### *Forma normal 1. Eliminar los grupos repetitivos*

La primera forma de normalización se centra en crear una tabla para cada conjunto de columnas relacionadas, y asignar a cada tabla una clave principal, o índice primario.

### *Forma normal 2. Eliminar datos redundantes*

La segunda forma establece que si una columna depende sólo en parte de una clave primaria, compuesta de dos o más atributos, ésta deberá colocarse en otra tabla.

### *Forma normal 3. Eliminar las columnas que no se relacionen con la clave principal*

---

<sup>2</sup> Curtis L. Smith y Michael C. Amundsen. Programación de Base de Datos con Visual Basic, Prentice Hall, México, 1999.

---

La tercera forma de normalización establece que si una columna no se ajusta a lo que describe la clave principal, deberá ser colocada en otra tabla.

*Forma normal 4. No almacenar datos calculados en las tablas*

La cuarta forma de normalización establece que no debe haber campos cuyo contenido sea el resultado de algún cálculo.

*Forma normal 5. Aislar las múltiples reglas dependientes*

La quinta y última forma de normalización establece que:

- No deberá existir tabla alguna que contenga más de una relación de uno a uno o de uno a muchos directamente establecida.
- Debe aislar las múltiples relaciones conexas dentro de las bases de datos.

## 1.2. Programación Orientada a Objetos (POO)

La POO es un *paradigma* de programación, que basa la construcción de sistemas de *software* en módulos obtenidos a partir de los objetos que manipula. Se define como un método de implementación en el que los programas están organizados como colecciones de objetos o clases, donde cada uno es un elemento de alguna clase, y donde todas las clases son miembros de una jerarquía de clases...<sup>3</sup>

Los módulos con los que se construye un programa en POO están basados en los datos del problema, y no en las acciones. Los datos se denominan objetos, y sus tipos se denominan clases.

### 1.2.1. Conceptos fundamentales de la POO

A continuación se procede a explicar los conceptos que conforman el mecanismo básico de la POO que son: objetos, mensajes, métodos, clases y subclasses.

#### *Objetos*

Un objeto es una encapsulación genérica de datos y procedimientos para manipularlos. Dicho de otra forma, un objeto es una entidad que tiene unos atributos particulares, los datos y unas formas de operar sobre ellos, los métodos. Por lo tanto, un objeto contiene por una parte, operaciones que definen su comportamiento y por otra variables, manipuladas por esas operaciones, que definen su estado.

---

<sup>3</sup> H. M. Deitel y P. J. Deitel. Como Programar en Java, Prentice Hall, España, 1998.



### *Mensajes*

Los mensajes son el medio a través del cual interactúan los objetos, la funcionalidad compleja y útil de un programa orientado a objetos no proviene de un sólo objeto, que no suele ser muy útil, sino de la interacción y comunicación entre varios objetos. Los objetos de software interactúan y se comunican entre ellos utilizando los mensajes. Los elementos que componen un mensaje en la POO son:

- El objeto donde se deposita el mensaje.
- El nombre del método que hay que realizar.
- Los parámetros que requiere el método.

### *Clases*

Una clase es la descripción generalizada que describe una colección de objetos similares, dicho de otra forma es el molde o esquema que describe una colección de valores llamados instancias. Una clase describe los métodos y atributos que definen las características comunes a todos los objetos de esa clase. Por definición entonces, todos los objetos que existen dentro de una clase reciben los atributos y las operaciones disponibles para la manipulación de los objetos de la misma clase.

### *Subclases*

Una característica primaria que define un sistema orientado a objetos es la manera en que trata las relaciones estructurales y *semánticas* entre clases de objetos y elimina la redundancia de almacenar el mismo atributo o método más veces de lo necesario.

Estas definiciones implican la existencia de una jerarquía de clases en la cual los atributos y operaciones de cualquier "clase" pueden ser heredados por subclases que pueden añadir, cada una de ellas, atributos y métodos.

## **1.2.2. Características de la POO**

Las características fundamentales de la programación orientada a objetos son: abstracción, encapsulamiento, herencia y polimorfismo.

### *Abstracción*

La abstracción, en la POO, consiste en la generalización conceptual de los atributos y propiedades de un determinado conjunto de objetos. La clave de la POO está en abstraer los métodos y los datos comunes a un conjunto de objetos y almacenarlos en una clase. Desde este nivel de abstracción, la introducción o

eliminación de un objeto en una determinada aplicación supondrá un trabajo mínimo o nulo.

### *Encapsulamiento*

El encapsulamiento u ocultamiento de información se refiere a la práctica de incluir dentro de un objeto todo lo que necesita, de tal forma que ningún otro objeto necesite conocer nunca la estructura interna. Esta característica permite ver un objeto como una caja negra, en la que se ha metido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna.

### *Herencia*

La herencia es un mecanismo para compartir automáticamente métodos y atributos entre clases y subclasses. Esta característica de la POO está fuertemente ligada a la reutilización de código, esto es, el código de cualquiera de las clases existentes puede ser utilizado sin más que crear una clase derivada de ella.

El concepto de herencia conduce a una estructura jerarquizada de clases, lo cual no significa que en POO todas las relaciones de entre clases deban ajustarse a una estructura jerárquica. La herencia puede también ser múltiple; esto es, una clase puede derivarse de dos o más clases.

### *Polimorfismo*

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando el mismo medio de acceso.

## **1.2.3. Ventajas de los Lenguajes Orientados a objetos**

Las tecnologías de objetos llevan a reutilizar, y la reutilización (de componentes de software) lleva a un desarrollo de software más rápido y a programas de mejor calidad. El software orientado a objetos es más fácil de mantener debido a su independencia funcional. Además, los sistemas orientados a objetos son más fáciles de adaptar y más fácilmente escalables.

Entre las múltiples ventajas con que cuenta los lenguajes de programación orientados a objetos podemos encontrar las siguientes:

- Los objetos bien diseñados son la base para sistemas que se construyen a partir de módulos reutilizables, dando lugar a una mayor productividad.

- La reutilización de las clases que han sido probadas en circunstancias reales, en proyectos anteriores, dan lugar a sistemas de mayor calidad y con menos errores.
- La herencia hace posible definir sistemas más fiables, más fáciles de ampliar y menos costosos de mantener.
- Un diseño orientado a objetos es una herramienta clave para enfrentarse a problemas complejos porque favorece el paso de pequeños o grandes sistemas.
- Las aplicaciones requieren escribir menos líneas de código, menos sentencias de bifurcación y por lo tanto, los módulos son más comprensibles.
- La encapsulación ayuda a construir sistemas más seguros.
- En contraste con la programación tradicional, donde siempre existe la posibilidad de que una modificación de un procedimiento o de una estructura de datos afecte a otros procedimientos, la programación orientada a objetos no necesita examinar todo el código fuente para ver si un cambio local afecta a otras partes.

Una clase es un tipo de datos abstracto<sup>4</sup>, es una noción matemática, sólo adecuada para la etapa de especificación, una clase describe una colección de valores llamados instancias. La clase es el molde, el esquema; un objeto es un valor particular de ese molde o esquema. Por lo que un objeto es una instancia en tiempo de ejecución de una clase.

## 1.3. Plataforma de Ambiente y Desarrollo

Existen diversas tecnologías mediante las cuales es posible la implementación de los sistemas de Información existentes, entre toda la gama de estas tecnologías encontramos las desarrolladas por Microsoft que en últimas fechas a propuesto una nueva plataforma de desarrollo sobre ambientes amigables y de alta compatibilidad con las tendencias tecnológicas de hoy en día.

### 1.3.1. .NET Framework<sup>5</sup>

*.NET Framework* constituye la plataforma y el elemento principal sobre el que se asienta la solución para el desarrollo de aplicaciones para las plataformas de sistemas operativos de Microsoft.

---

<sup>4</sup> Ver Sección 1.2.2.

<sup>5</sup> Luis Miguel Blanco. Programación en Visual Basic .NET. Grupo EIDOS, España, 2003.

De cara al programador, es la pieza fundamental de todo un nuevo modelo de trabajo, ya que proporciona las herramientas y servicios que necesitará en su labor habitual de desarrollo. En a figura 1.1. se muestra un diagrama con la distribución de elementos dentro del entorno de *.NET Framework*.

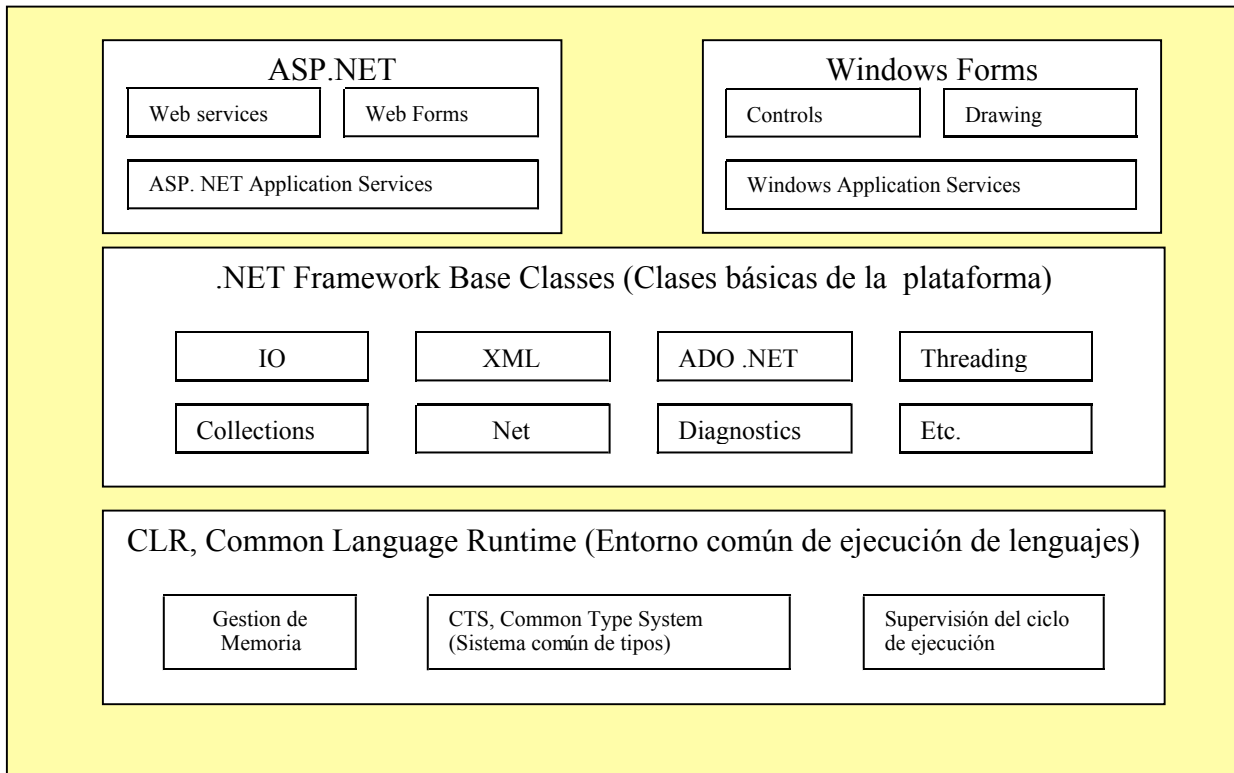


Figura 1.1 Esquema de componentes dentro de la plataforma *.NET Framework*.

*.NET Framework* permite el desarrollo de aplicaciones a través del uso de un conjunto de herramientas y servicios que proporciona y que pueden agruparse en tres bloques principales: el entorno de ejecución común o *Common Language Runtime* (CLR a partir de ahora); la jerarquía de clases básicas de la plataforma o *.NET Framework Base Classes*; y el motor de generación de interfaz de usuario, que permite crear interfaces para la Web o para el tradicional entorno a Windows, así como servicios para ambos entornos operativos.

En la base del entorno de ejecución se encuentra el CLR, que constituye el núcleo de *.NET Framework*, encargándose de la gestión del código en cuanto a su carga, ejecución, manipulación de memoria, seguridad, etcétera.

En el nivel intermedio, se sitúa la jerarquía de clases básicas del entorno de ejecución, que constituyen un sólido grupo de librerías de servicios a disposición del

programador, para multitud de tareas como, gestión del sistema de archivos, manipulación multihebra, acceso a datos, etcétera.

Entre el grupo de servicios cabe destacar el nuevo motor de acceso a datos llamado ADO.NET (*Activex data Object .NET*), mediante el cual nos basaremos para la conectividad con la base de datos del sistema. ADO .NET es el modelo de objetos para el acceso a datos incluido en la jerarquía de clases de la plataforma *.NET Framework*. Se trata de una evolución de ADO (*Activex Data Objects*), el anterior modelo para la gestión de datos, incluido en versiones anteriores de Visual Studio.

En ADO, la representación de datos en memoria es el conjunto de registros; mientras que en ADO.NET, es el conjunto de datos. Existen importantes diferencias entre ellos, entre las que podemos mencionar:

- En ADO, la búsqueda de las filas del conjunto de registros se realiza de forma secuencial; mientras que en ADO.NET, las filas se representan como colecciones
- En ADO.NET las conexiones se abren sólo el tiempo necesario para realizar una operación de base de datos, como una selección o una actualización, ya que es posible incorporar las filas en un conjunto de datos y a continuación trabajar con ellas sin estar conectado con el origen de datos. En ADO, el conjunto de registros puede proporcionar acceso desconectado, pero se diseñó principalmente para el acceso conectado.
- ADO .NET ha sido ampliado para cubrir todas las necesidades que ADO no ofrecía. Está diseñado para trabajar como formato universal con el lenguaje de intercambio de marcas XML (*eXchange Marcus Language*), que es ya un estándar para el formato de transmisión de datos.

Finalmente, en el nivel superior de la figura 1.1. encontramos las clases que permiten el diseño del interfaz de usuario de nuestras aplicaciones. Ya sea si necesitamos desarrollar aplicaciones para Internet o para un entorno dentro del sistema operativo.

En el caso del desarrollo de aplicaciones para Internet la plataforma proporciona como herramienta el uso de ASP .NET (*Activex Server Page .NET*), que otorga los servicios necesarios para la programación dentro de este ambiente de Red.

Para el desarrollo dentro del entorno del sistema operativo se proporciona *Windows Forms*, que es la nueva generación de formularios, que poseen características avanzadas y muy superiores a los conocidos en versiones anteriores. El Diseñador de Windows Forms proporciona una solución rápida de desarrollo para la creación de aplicaciones para Windows.

Adicionalmente, existe la posibilidad de que necesitemos servicios del sistema que no requieran interfaz del usuario en absoluto. Este aspecto también está

---

contemplado por la plataforma, permitiéndonos, por ejemplo, la creación de servicios para Windows 2000 y NT.

### 1.3.2 Visual Basic .NET<sup>6</sup>

Con la herencia visual, es posible simplificar enormemente la creación de aplicaciones basadas en Windows, centralizando la interfaz de usuario y la lógica común de toda la solución en formularios primarios.

Visual Basic ha sido actualizado para incluir muchas nuevas y mejores características, que hacen de éste un poderoso lenguaje orientado a objetos, estas características incluyen:

- Herencia. Visual Basic .NET soporta herencia por medio de la definición de clases, que sirven como base para la creación de subclases que pueden extender las propiedades y sus métodos de su clase padre.
- Manejo de Excepción Estructurada. Visual Basic .NET usa una versión avanzada del *try...catch...finally*, sintaxis soportada por otros lenguajes (C++, JAVA). La excepción estructurada combina un control moderno estructurado, con excepciones, protección de bloques y filtrado.
- Sobrecarga. Habilita la definición de propiedades, métodos que tienen el mismo nombre pero que manejan distintos tipos de datos. Los procedimientos sobrecargados proveen la implementación de un solo procedimiento con distintos tipos de resultados.
- Permite la sobrescritura de propiedades y métodos. Esta característica permite la reescritura de las características heredadas de los objetos padre. Los miembros sobrescritos tienen los mismos argumentos que los miembros heredados de la clase base, pero diferente implementación.
- Declaración de constructores y destructores. Permite el control de la inicialización de nuevas instancias de clase, así como los correspondientes métodos de destrucción para la liberación de recursos de sistema.
- Tipos de datos. Visual Basic .NET introduce árboles de nuevos tipos de datos para el manejo de tipos distintos en un solo tipo.
- Interfaces. Permite la implementación de la descripción de las propiedades y los métodos de las clases, así como la creación de ligas virtuales para implementación de estos.

---

<sup>6</sup> Microsoft Corporation. Manual de Referencia de Visual Basic .NET. Microsoft Press, E.U., 2002.

- Miembros Compartidos. Los miembros compartidos son propiedades, procedimientos y campos que son compartidos por todas las instancias de una clase.
- Referencias. Permiten el uso de objetos en otros ensamblados de código.
- Espacios de Nombre. Esta característica previene conflictos de declaración para organizar clases, interfaces y métodos dentro de jerarquías.
- Ensamblados. Los ensamblados reemplazan y extienden las capacidades de las librerías de tipos, describiendo todos los archivos requeridos para una aplicación o componente. En particular un ensamblado puede contener uno o más espacios de nombre.
- Atributos. Los atributos proveen información adicional acerca de los elementos del programa.
- Multiprocesamiento. Visual Basic soporta programación multihilo que es la habilidad de asignar tareas individuales a procesos separados en hilos.

Visual Basic .NET es una jerarquía de clases que están incluidas en el Framework.NET. Con Visual Basic .NET, Visual Basic sigue siendo la herramienta más productiva para la creación de aplicaciones que se ejecutan en el sistema operativo Microsoft Windows. Con Visual Basic .NET, se puede seguir aprovechando los conocimientos y capacidades para crear aplicaciones y servicios *Web XML*.

### **1.3.3. Arquitecturas de aplicación**

La arquitectura del sistema es una importante característica en el diseño de aplicaciones, puesto que ésta define la forma en que los elementos de la aplicación interactúan y la funcionalidad que cada elemento provee.

Una aplicación típica que interactúa con un usuario consiste de tres elementos: presentación, lógica de la aplicación y servicio de datos. Un servicio es una unidad de la lógica de la aplicación que implementa operaciones, funciones, o transformaciones que son aplicadas a objetos. Cada uno de estos elementos (o servicios) poseen ciertos atributos que se muestran en la Tabla 1.1.

#### *Presentación*

Es también conocida como la interfaz de usuario (UI), enfocada a la interacción con el usuario.

#### *Lógica de la aplicación*

También definida como reglas del negocio, calcula y determina el flujo de la aplicación. Las reglas de negocio son restricciones, usualmente impuestas, que las compañías usan como ayuda para operar en negocios particulares, Las reglas de negocio definen una línea base de los requerimientos de la aplicación y proveen una guía para su desarrollo. En términos prácticos, las reglas de negocio son las claves que los desarrolladores usan para la implementación de la lógica de la aplicación.

### *Servicio de Datos*

Maneja la información almacenándola en forma de datos y proveyendo la funcionalidad de la relación entre los mismos, es fundamentalmente la base de datos de la aplicación.

<b>Tipo de Servicio</b>	<b>Atributo del Servicio</b>
Presentación	Presentación de información y funcionalidad, navegación, y protección de la interfaz de usuario consistencia e integridad.
Lógica de la aplicación	Compartir políticas de negocios, generación de información de negocios de datos, y protección de la integridad de negocios.
Servicio de Datos	Definición de Datos, almacenamiento y retiro de datos persistentes, y protección de integridad de datos.

*Tabla 1.1. Atributos de los tipos de servicios.*

## **Tipos de Arquitecturas**

Existen tres tipos de Arquitecturas de Sistemas (o aplicaciones), que se clasifican con base en las capas (una capa es un concepto lógico que define la separación entre los elementos de una aplicación) y son: capa-simple (o monolítica), dos-capas y multicapa.

### *Aplicaciones de Capa-Simple*

En una aplicación de capa-simple, sólo una capa soporta la presentación, la lógica de la aplicación y el servicio de datos. Los datos pueden ser físicamente almacenados en cualquier localización, como en un servidor. Mientras que la lógica de la aplicación y la funcionalidad para el acceso a datos es parte de la aplicación



### *Aplicaciones de Dos-Capas*

Aplicaciones de Dos-Capas o Cliente/Servidor, agrupa la presentación y los componentes de la lógica de la aplicación en la maquina del cliente y acceda a la fuente de datos usando una conexión de red. En una aplicación de dos capas, la interfase de usuario y reglas de negocio son una capa simple que se ejecuta en la computadora del cliente. Una aplicación por separado, como SQL Server o Oracle, proveen el servicio de datos.

### *Aplicaciones Multicapa*

Las Aplicaciones Multicapa pueden ser implementadas usando múltiples computadoras sobre una red. Este tipo de aplicaciones es conocido como aplicaciones distribuidas, o aplicaciones de n-capas. Una aplicación de n-capas representa una instancia especial de una aplicación de tres-capas en el que una o más de las capas son separadas dentro de adicionales capas, que proveen gran escalabilidad. Las tres capas son generalmente descritas por un usuario (primero), negocio (segundo o medio) y datos (tercero).

En arquitecturas multi-capa, la presentación, la lógica de la aplicación y los datos son elementos conceptualmente separados. Estas capas no necesitan corresponder a una localización física en una red. En las aplicaciones multicapa, el cliente provee solo una capa, la interfaz del usuario. Las reglas del negocio son desarrolladas en un sistema entre la interfase del usuario y el sistema de almacenamiento de datos.

### **1.3.4. SQL Server**

SQL Server es un manejador de base de datos desarrollado por Microsoft que puede ser ejecutada en cualquier versión de Windows a partir de de la versión de Windows 98. Como sistema Manejador de Base de Datos proporciona interfaces para la creación y administración de la información, que son muy sencillas de manejar, lo que da oportunidad a los desarrolladores de optimizar tiempos. Su objetivo principal es el ser un ambiente en el que sea posible guardar y recuperar información de la Base de Datos en forma eficiente.

Dentro de las funciones de SQL Server se encuentran:

- Almacena, recupera y modifica datos.
- Guarda la consistencia de los datos.

- Soluciona problemas de concurrencia.

Una vez definidos los conceptos que usaremos para el desarrollo del sistema administrador de información, comenzaremos la etapa de análisis que, en forma general, comprenderá la descripción de las necesidades y requerimientos del sistema.

# 2

## Análisis del Sistema

---

En este capítulo hacemos una revisión de la situación actual en la que se encuentra la Institución Educativa, mostrando un panorama general de sus principales actividades. Posteriormente definimos la problemática que presenta la institución en la administración de la información, basándonos en los procedimientos que actualmente utilizan para llevar a cabo sus actividades. Finalmente, planteamos la propuesta de solución y los alcances del proyecto.

El análisis y la especificación de requisitos son un conjunto de actividades que desarrollan el analista de sistemas, el cliente y los usuarios, con el propósito de identificar y recopilar la descripción y las características operacionales del sistema (manejo de información, reglas del negocio, etc.).

El proceso del análisis consiste en obtener información completa y clara sobre el objetivo, datos y funciones, de la institución en cuestión. Estos requisitos deben ser revisados para conseguir junto con el usuario claridad, completitud y consistencia en la información. Ésta es obtenida a partir de documentos, manuales y principalmente de entrevistas entre el analista, el cliente y los usuarios finales, en estas entrevistas el cliente debe presentar una visión y descripción general del sistema y de las funciones que espera que éste realice, así como las características de las interfaces.

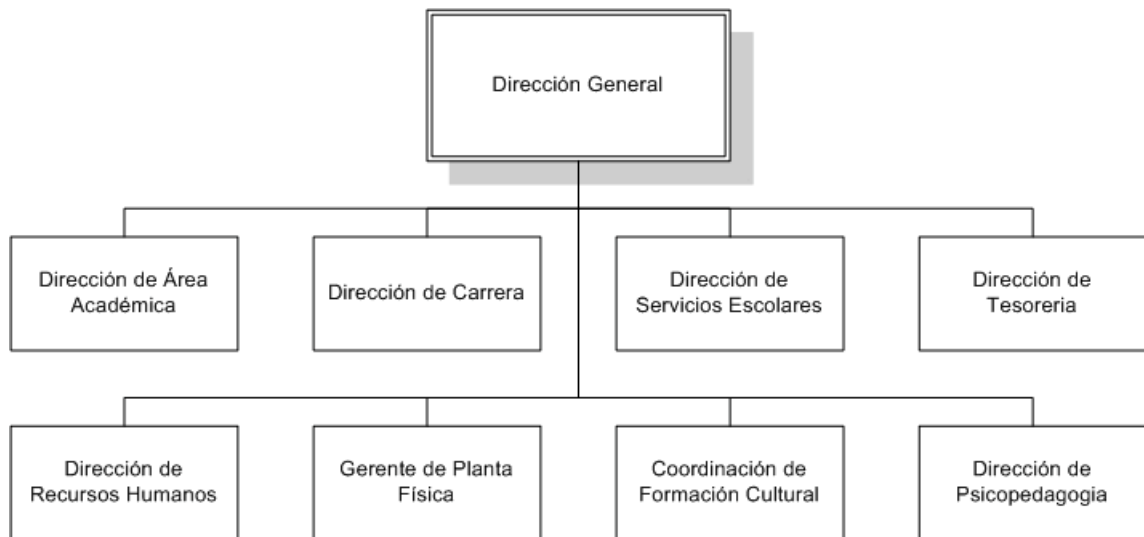
Durante los últimos años, las tendencias en sistemas de información en las organizaciones de gran tamaño se han dirigido a garantizar los servicios y herramientas, para que sus miembros y usuarios desarrollen sus actividades de forma óptima.

## 2.1. Situación actual

Para lograr un diseño y desarrollo adecuado de un sistema, es importante la determinación de la situación actual en el manejo de la información de la institución. Y esto por medio de la investigación a través de documentos y entrevistas con el personal administrativo y de los procesos que manejan las diferentes áreas involucradas. Con base en esto se encontrarán las deficiencias y requisitos, se plantearán los objetivos y los alcances del sistema.

La Institución Educativa, para la cual se desarrollará el proyecto, tiene como objetivo preparar profesionales competitivos cuya formación académica, valores y principios les permitan contribuir al desarrollo de la sociedad, afrontando con éxito los retos que ofrece el mundo actual, generando conocimiento y reafirmando día con día su presencia Institucional.

La institución se encuentra dividida en Coordinaciones y Direcciones de acuerdo al diagrama de la figura 2.1:



*Figura 2.1. División de Direcciones y Coordinaciones.*

Las actividades principales que cumplen cada una de estas áreas son (cabe comentar que aquí profundizamos en aquéllas que son de mayor importancia en el desarrollo del sistema):

- *Dirección General.* Se encarga de supervisar el funcionamiento general de la institución y tomar decisiones relacionadas con el patrimonio de la misma.
- *Direcciones de Área Académica.* Supervisar el buen funcionamiento y la administración del programa académico en el nivel de ciencias básicas y materias de humanidades, atender a los alumnos que lo cursen, contratar y coordinar a los profesores para que impartan sus cursos con el nivel profesional y docente que se requiere. Fomentar un espíritu universitario y mantener los contenidos y la metodología de los programas en constante cuestionamiento y desarrollo para asegurar un apoyo efectivo a las Áreas Académicas.
- *Direcciones de Carrera.* Es la responsable del desarrollo integral y del bienestar de los miembros de la carrera, así como del cumplimiento y mejora del programa académico que se imparte. Asimismo, coordina todos los aspectos relacionados con la preparación técnico profesional del estudiante, mediante una esmerada atención académica de su profesorado, el fomento de los valores humanos universales, inspirando en los alumnos y personal docente un espíritu de servicio, conciencia social y apertura a la cultura. En lo que respecta a las Direcciones de Área Académica y de Carrera, sus funciones son similares, manejándose cada una sobre su ámbito; para la Dirección de Área Académica en el nivel de ciencias básicas y humanidades, y la Dirección de Carrera, enfocado a materias especializadas en el ramo, de las cuales podemos mencionar:
  - Notificar al personal docente y administrativo a su cargo la promoción o cambio respectivo.
  - Conducir las actividades de reclutamiento y selección del personal docente y personal no académico de su área.
  - Elaborar los horarios de clase de su área y atender cualquier asunto académico que se presente durante el periodo de inscripciones.
  - Corregir las listas de asistencia y entregarlas a tiempo a su planta docente.
  - Asegurarse que cada profesor de nuevo ingreso tenga una categoría y explicarles los criterios de clasificación.
  - Enterar oportunamente a la Dirección de Tesorería las modificaciones en la retribución del personal docente y administrativo a su cargo.
  - Detectar las necesidades de capacitación y actualización de su personal docente y personal administrativo a su cargo.
  - Asegurar que el expediente físico de su planta docente esté siempre actualizado de acuerdo con los periodos estipulados.

- Registrar los incidentes críticos y documentar e informar las faltas graves del personal docente y personal administrativo a su cargo, para tomar una decisión oportuna.
- Revisar, desarrollar y seleccionar material didáctico y textos apropiados para el desempeño exitoso de los programas académicos.
- *Dirección de Servicios Escolares.* Se encarga de dirigir y evaluar la planeación, organización y desarrollo de los procesos de ingreso, control y egreso de los alumnos de la Universidad, así como dirigir y evaluar las actividades de otorgamiento, control y seguimiento de la incorporación de estudios de escuelas de nivel superior, entre sus funciones específicas podemos encontrar:
  - Supervisar la consistencia de la información relacionada con el Programa de Control Escolar, durante todo el ciclo escolar y cursos de regularización.
  - Supervisar la solicitud y control de la documentación de los alumnos de nuevo ingreso.
  - Supervisar la asignación de matrícula a los alumnos de nuevo ingreso.
  - Realizar los trámites para validar ante la Secretaría de Educación Pública (SEP) el plan de estudio y los rediseños necesarios.
  - Concentrar y distribuir la documentación necesaria relativa al proceso de inscripción.
  - Efectuar los trámites legales para dar de baja al alumno.
  - Realizar los trámites necesarios de legalización para entregar los certificados y el título o grado académico, junto con la documentación del alumno, en tiempo y forma.
  - Expedir la constancia que certifica que el alumno está en posibilidades de iniciar su servicio social.
  - Supervisar la impresión y entrega oportuna a las direcciones correspondientes de las listas de asistencia, al inicio del ciclo escolar y en los periodos parciales subsecuentes.
  - Colaborar en el análisis y elaboración del reglamento general y los reglamentos internos, a fin de que estén en concordancia con los lineamientos generales de la SEP.

- Mantener contacto directo con las áreas docentes y de servicios, para el intercambio de la información necesaria para el logro de los objetivos institucionales.
  - Firmar, supervisar y controlar la expedición de documentos oficiales, como: constancias de estudio y servicio social; equivalencias y revalidaciones; certificados parciales y totales; títulos, diplomas y menciones honoríficas.
  - Supervisar y controlar el proceso de solicitud y acreditación de exámenes recepcionales y de grado, así como el registro de los libros de actas.
  - Supervisar que las boletas de calificación sean elaboradas inmediatamente después de cada periodo de exámenes.
  - Supervisar la atención e información dada por el personal de apoyo a los alumnos y padres de familia, e intervenir en aquellos asuntos que no pueden ser resueltos en esta primera instancia.
  - Supervisar, controlar y custodiar los expedientes de los alumnos, a fin de que su documentación esté debidamente integrada y cumpla con los requisitos legales de la SEP.
  - Elaborar los calendarios para cada ciclo escolar, así como los instructivos y formatos necesarios para orientar a los alumnos en sus trámites administrativos.
- *Dirección de Tesorería.* Llevar el registro contable y fiscal de las recursos financieros de la Institución, a fin de contar con la información necesaria para tomar decisiones y para cumplir puntualmente con las obligaciones hacendarías y laborales, algunas de sus funciones específicas son:
    - Monitorear en el transcurso del ciclo escolar, las colegiaturas no cubiertas, para dar aviso oportunamente a las Direcciones correspondientes a fin evitar un retraso mayor.
    - Entregar a los directores diariamente la relación de alumnos con adeudos pendientes, durante el periodo de inscripciones.
    - Realizar los trámites necesarios en caso de remoción o separación voluntaria del personal docente o administrativo.
    - Supervisar las requisiciones autorizadas y en su caso, programar los pagos.

- Supervisar y certificar que el alumno no tiene adeudos para poder entregarle los documentos oficiales. Coordinar el trabajo del equipo del personal a su cargo durante el periodo de inscripciones.
- Asesorar y apoyar a las autoridades universitarias sobre los aspectos de administración de recursos financieros.
- *Dirección de Recursos Humanos.* Realiza los procesos de administración del personal administrativo y de mantenimiento.
- *Gerente de Planta Física.* Coordinar y optimizar el uso de la infraestructura física de la institución, de manera que la comunidad universitaria desarrolle sus actividades académicas y extraacadémicas en un ambiente agradable, seguro, limpio, sano y armónico.
- *Coordinación de Formación Cultural.* Fomentar la capacidad de apreciación del arte y la cultura para contribuir al desarrollo integral de los alumnos y de la comunidad universitaria.
- *Direcciones de Desarrollo Psicopedagógico.* Ayudar a la comunidad universitaria a continuar con éxito su desempeño académico a través del desarrollo y fortalecimiento de habilidades de aprendizaje, técnicas de estudio, orientación profesional y consejería personalizada, y así evitar en lo posible la deserción escolar; asimismo, planear, aplicar y calificar el examen de admisión de los candidatos de nuevo ingreso y entregar los resultados.

Las carreras de licenciatura que se imparten dentro de la institución son: administración, contaduría, derecho, economía e informática.

Actualmente se cuenta con un manual de procedimientos muy apegado a la consecución y seguimiento de los trámites administrativos básicos, para el funcionamiento adecuado de la institución, entre estos procedimientos podemos detallar principalmente:

- Procedimiento para inscripción de alumnos de nuevo ingreso
- Procedimiento para inscripción de alumnos regulares.
- Procedimiento para inscripción de alumnos irregulares.
- Procedimiento para integrar el expediente del personal docente.
- Procedimiento para diseñar planes de estudio.
- Procedimiento para diseñar y rediseñar programas de materia.



## **Procedimiento para inscribir a los alumnos de nuevo ingreso**

### *Propósito*

Contar con una metodología que le permita a la Universidad inscribir a sus alumnos de manera rápida y eficiente.

### *Procedimiento*

En este procedimiento se involucran las Direcciones de Servicio Escolar, de Carrera, Tesorería y se crea un equipo de firmas, el cual está integrado por personal administrativo y coordinadores de carrera. El procedimiento se explica con base en el diagrama de flujo 2.2.

#### *A. La Dirección de Servicios Escolares:*

1. Se verifica que el alumno de nuevo ingreso aparezca en la “relación de alumnos aprobados en el examen de admisión”.
2. Se solicita a cada uno de los alumnos de nuevo ingreso la documentación respectiva y se verifica la recepción a través de los formatos de recibo de documentos donde se señala con una paloma aquéllos que sí se entreguen en ese momento. La documentación es la siguiente:
  - Acta de nacimiento en original y copia.
  - Certificado de secundaria y preparatoria en original y copia, o provisionalmente una constancia de estudios.
  - Copia de la Clave Única de Registro de Población (CURP).
  - 6 fotografías blanco y negro tamaño infantil en papel delgado.
3. Se entrega una copia del formato de recibo de documentos al alumno de nuevo ingreso y archiva la original para control interno.
4. Se verifica si la solicitud de inscripción tiene un número de matrícula, lo que significa que el alumno de nuevo ingreso está preinscrito. Si no tiene número de matrícula, debe asignárselo.
5. Se entrega la solicitud de nuevo ingreso y documentación al alumno de nuevo ingreso para que lo presente en el siguiente paso.
6. Se le indica al alumno de nuevo ingreso que se dirija al punto B para continuar el procedimiento.

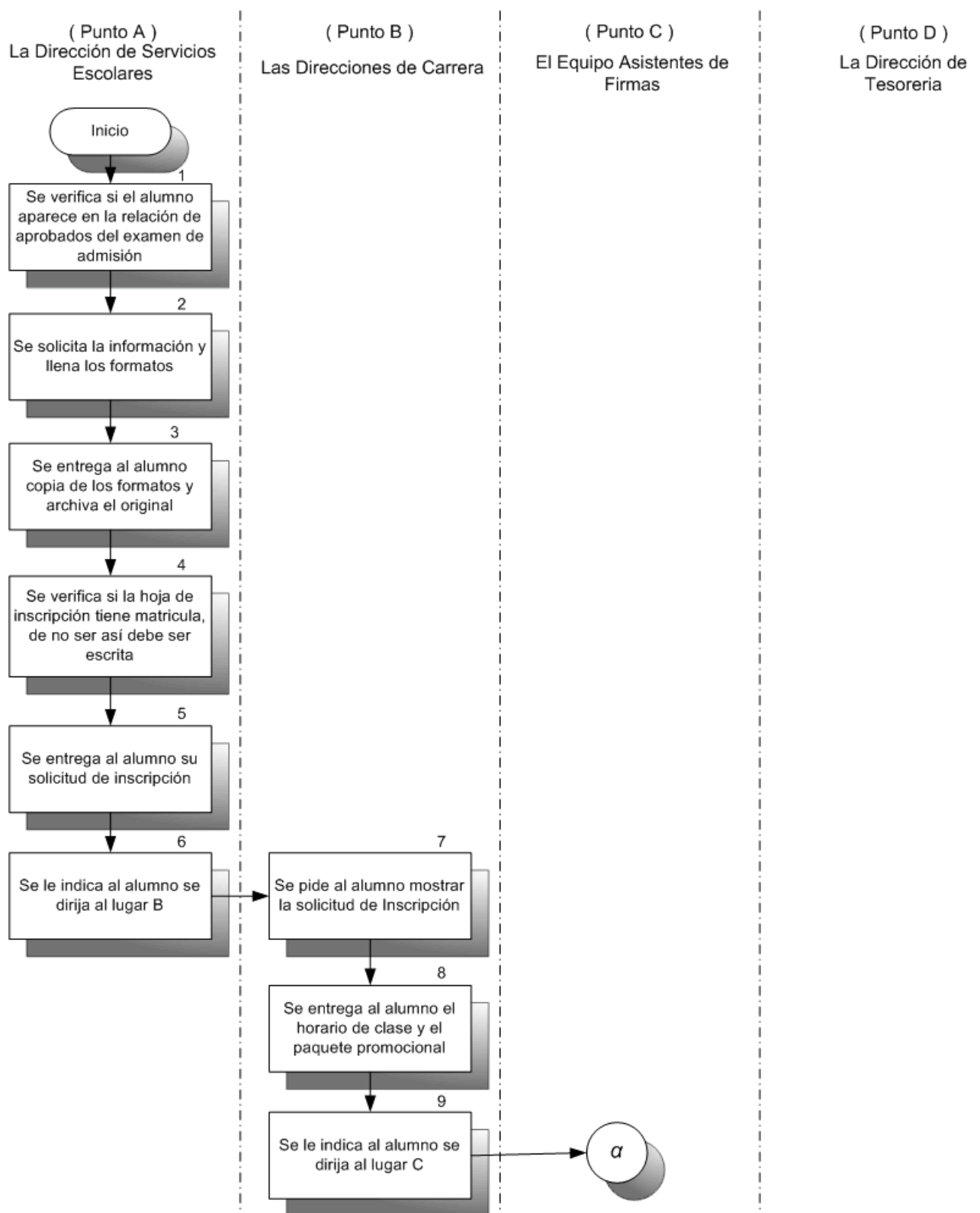


Figura 2.2. Procedimiento para inscribir a los alumnos de nuevo ingreso (Parte 1 de 3).

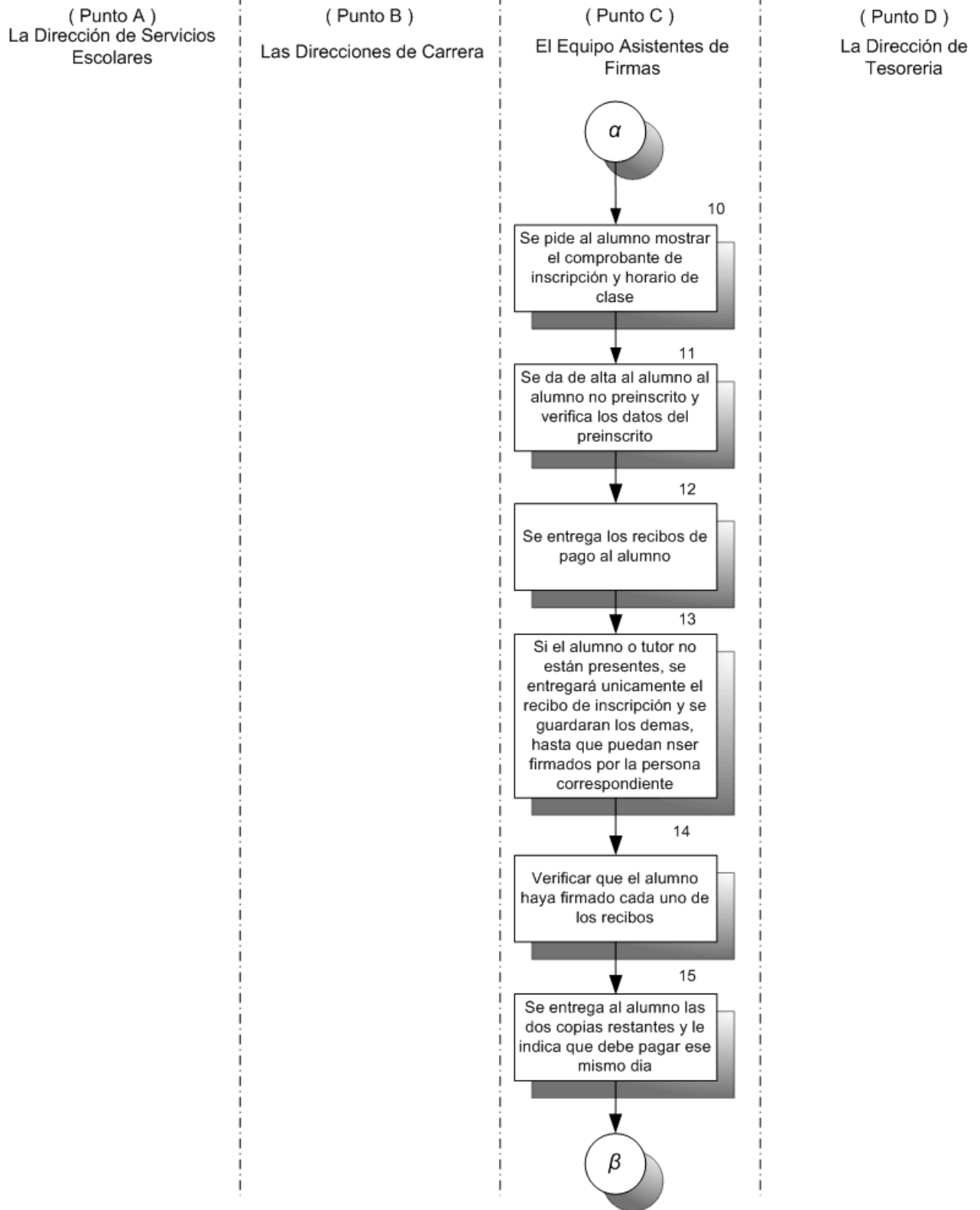


Figura 2.2. Procedimiento para inscribir a los alumnos de nuevo ingreso (Parte 2 de 3).

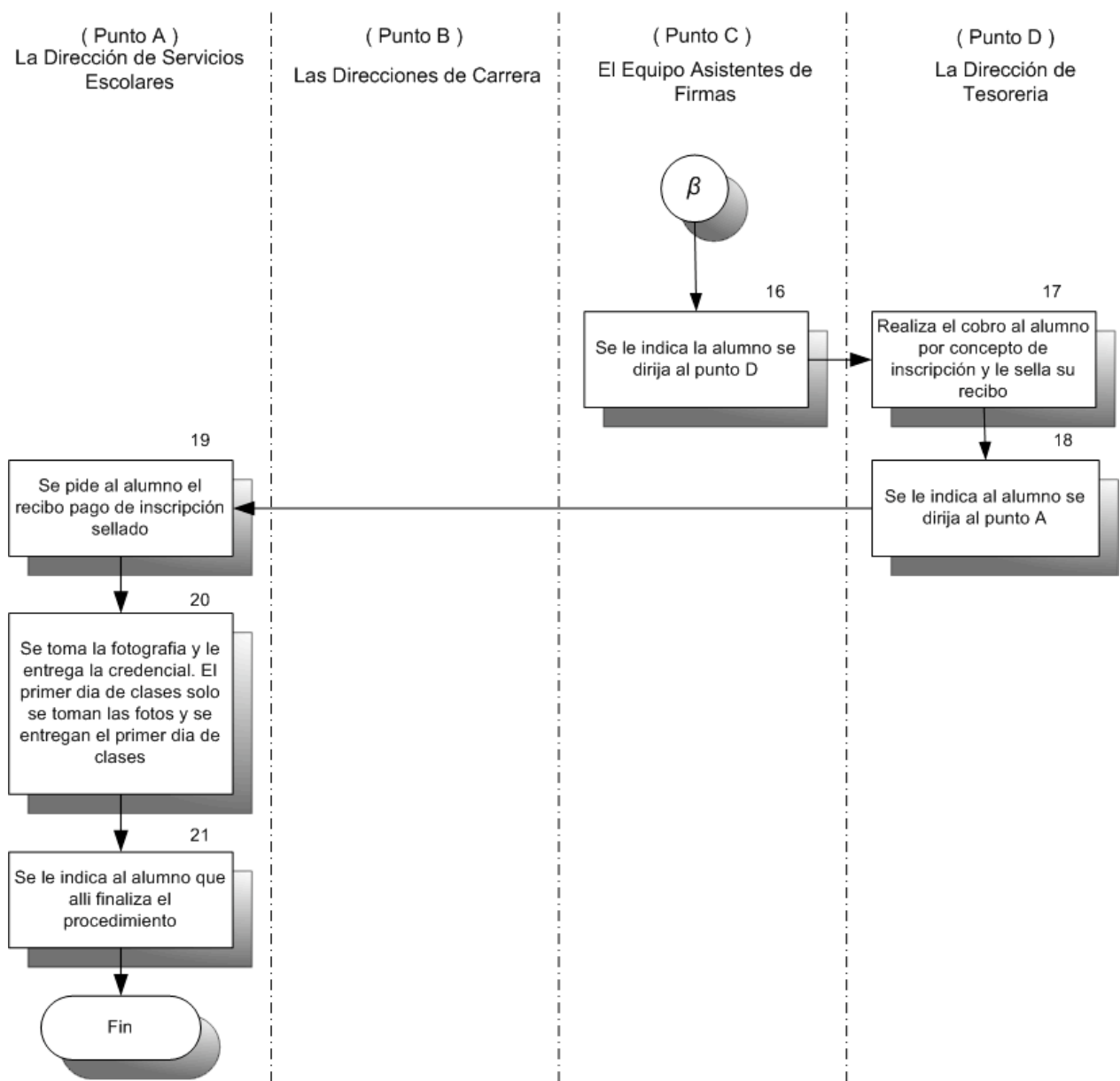


Figura 2.2. Procedimiento para inscribir a los alumnos de nuevo ingreso (Parte 3 de 3).

### B. Las Direcciones de carrera: (Punto B)

7. Se muestra solicitud de inscripción como requisito para continuar el procedimiento de inscripción.
8. Se entrega al alumno de nuevo ingreso el horario de clase correspondiente al alumno de acuerdo al avance de créditos de su vida académica.
9. Se le indica al alumno pase al punto C para continuar el procedimiento.

*C. El equipo de asistentes de firma: (Punto C)*

10. Se recibe la solicitud de inscripción y se revisa el horario de clases del alumno como requisito para continuar el procedimiento.
11. Se da de alta al alumno de nuevo ingreso basándose en la información contenida en la solicitud de inscripción que el alumno de nuevo ingreso debe entregar.

- Datos del Alumno

- Matrícula: Número consecutivo asignado en la solicitud de inscripción.
- Especialidad: Estudios que cursará.
- Nombre: Apellidos\_ :Nombre(s)
- Fecha de nacimiento: Con dos dígitos en: mes/día/año
- Sexo: Masculino o Femenino.
- Domicilio: Calle y número (sin #).
- Colonia: Nombre de la colonia.
- Ciudad: Ciudad, \_Abreviatura del estado.
- Teléfono: Local X-XX-XX o foráneo (Lada) X-XX-XX.
- CP: Número de Código Postal.
- Semestre: Número del semestre que cursará.
- Datos del Padre o Tutor
- Título: Abreviatura de la profesión del tutor.
- Nombre: Nombre completo del tutor, empezando por su nombre de pila.
- Domicilio: Calle y número (sin el símbolo #) donde vive el tutor.
- Colonia: Nombre de la colonia.
- Ciudad: Ciudad, Abreviatura del estado.
- Teléfono: Teléfono donde se puede localizar al tutor (casa y oficina).
- CP: Número de código postal.

12. Se entrega los recibos de pago al alumno de nuevo ingreso, los cuales deben contener la siguiente información:
  - Matrícula: Localiza el número de la matrícula del alumno.
  - Concepto: Debe decir "COLEGIATURAS".
  - Inscripción: Se escribe la cuota de inscripción anual.
  - Recibos: Normalmente es 12 (anual), excepcionalmente con autorización por escrito puede ser 6 (semestral) ó 1 (de contado).
  - Moratorio: 5%.
  - Fecha de vencimiento: del año en curso.
  - Parcialidad: Se escribe el monto de la colegiatura mensual.
13. El alumno firma cada uno de los recibos de pago. Si el alumno o tutor no están presentes, se entregará únicamente el recibo de inscripción y se guardarán los demás, hasta que puedan ser firmados por la persona correspondiente.
14. Se verifica que el alumno haya firmado cada uno de los recibos de pago.
15. El personal desprende los recibos de pago originales y se queda con ellos.
16. Se entrega al alumno las dos copias restantes y pedir al alumno pase pagar el recibo de inscripción, ya que de lo contrario no podrá ser dado de alta.
17. Se le indica al alumno de nuevo ingreso que se dirija al punto E para continuar el procedimiento.

*E. La Dirección de Tesorería: (Punto E)*

18. El personal de ventanilla cobra al alumno de nuevo ingreso la inscripción y sella el recibo de pago con la leyenda "PAGADO" y con la fecha del día de pago. En caso de que el alumno solicite una factura se le pide que pase por ella en la primera semana de clases. Se entrega al alumno el recibo de pago de la inscripción sellado con la leyenda "PAGADO" y con la fecha de pago.
19. Se le indica al alumno de nuevo ingreso se dirija al punto F para continuar el procedimiento.

*F. Las Direcciones Académicas: (Punto F)*

20. Se solicita al alumno el recibo de pago de la inscripción sellado con la leyenda "PAGADO" y la fecha de pago.

21. Se anota el número de matrícula del alumno para llevar un control interno.
22. Se toma la fotografía y se entrega la credencial al alumno. El primer día de inscripciones sólo se toman las fotos y se entregan el primer día de clases.
23. Se le informa al alumno de nuevo ingreso que allí finaliza el procedimiento de inscripción.

## **Procedimiento para inscribir a los alumnos regulares de reingreso**

### *Propósito*

Este procedimiento abarca a los alumnos regulares de reingreso de las carreras profesionales, desde que llegan a la Universidad para hacer sus trámites de inscripción hasta que se les actualizan sus datos en el sistema.

### *Procedimiento*

En este procedimiento se involucran las Direcciones de Servicio Escolar y Tesorería, y se crea un equipo de firmas, el cual está integrado por personal administrativo y coordinadores de carrera. El procedimiento se explica con base en el diagrama de flujo 2.3.

#### *A. El equipo de asistentes de firma: (Ubicación lugar D)*

1. Se verifica que la matrícula del alumno no aparezca en la “Relación de alumnos con adeudos pendientes”. Si aparece su matrícula en ese listado se informa al alumno que no puede inscribirse en ese momento sino hasta después del periodo de inscripciones, una vez que esté al corriente en sus pagos. Si no aparece su matrícula en el listado puede continuar el procedimiento.
2. Se entrega al alumno los recibos de pago de colegiaturas. Si el alumno o tutor no están presentes, se entregará únicamente el recibo de inscripción y se guardarán los demás, hasta que puedan ser firmados por la persona correspondiente.
3. Se verifica que el alumno haya firmado cada uno de los recibos de pago.
4. El personal que atiende desprende los recibos de pago originales y se queda con ellos.
5. Se entrega al alumno las dos copias restantes y se le pide al alumno pase pagar el recibo de inscripción, ya que de lo contrario no podrá ser dado de alta. Se le indica al alumno que se dirija al lugar B para continuar el procedimiento.

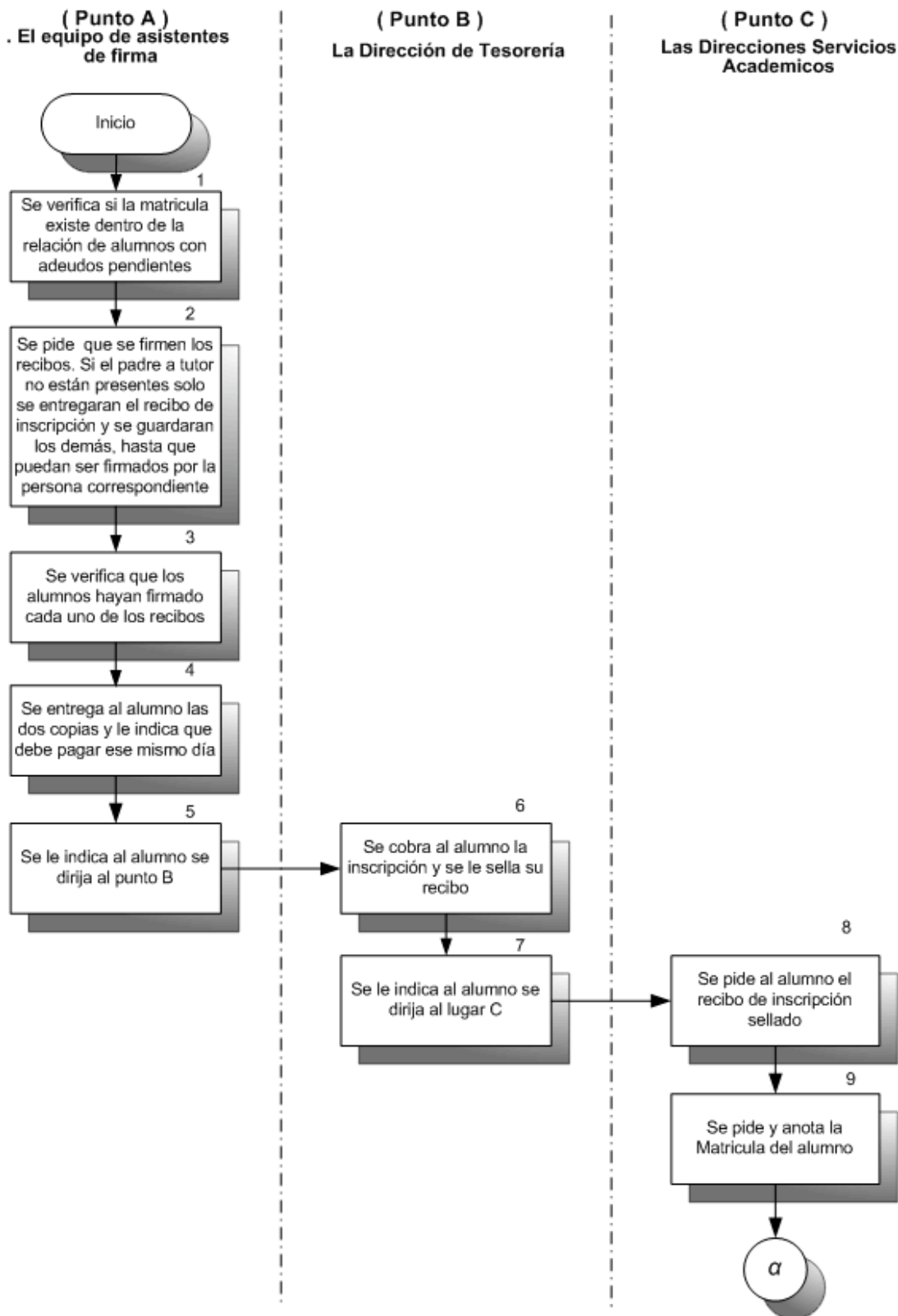


Figura 2.3. Procedimiento de reinscripción de alumnos regulares (1 de 2).



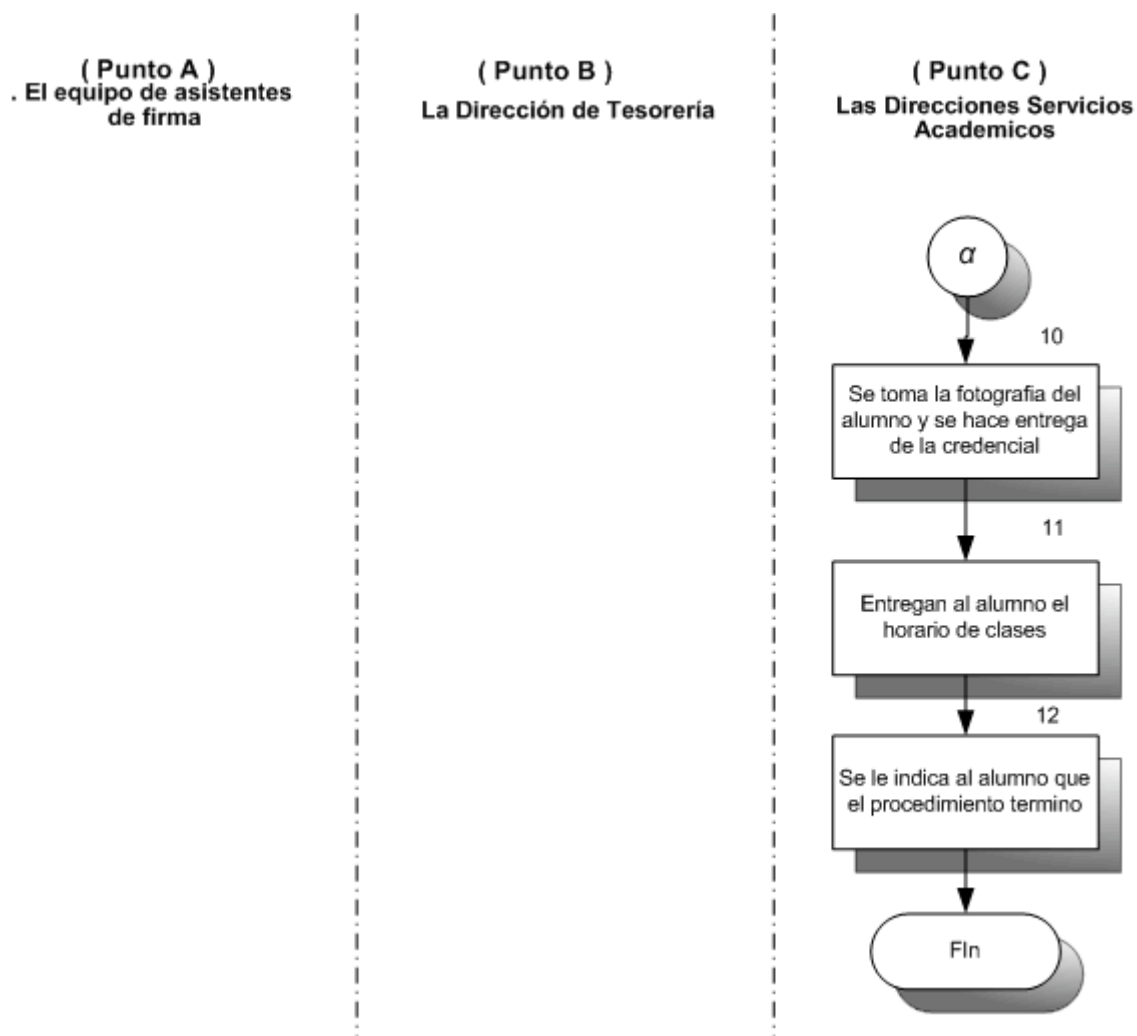


Figura 2.3. Procedimiento de reinscripción de alumnos regulares (2 de 2).

**B. La Dirección de Tesorería: (Punto B)**

6. Realiza el cobro al alumno regular de reingreso por concepto de inscripción y sella el recibo de pago con la leyenda “PAGADO” y con la fecha del día de pago. En caso de que el alumno solicite una factura debe pasar por ella en la primera semana de clases.
7. Se le indica al alumno que pase al punto C para continuar el procedimiento.

**C. La Dirección Servicios Académicos: (Punto C)**

8. Se le pide al alumno el recibo de pago de la inscripción sellado con la leyenda “PAGADO” y la fecha de pago.

9. Se le pide y anota el número de matrícula del alumno para llevar un control interno.
10. Se le toma la fotografía y se le hace entrega de la credencial al alumno.
11. Se le entrega al alumno el horario de clase.
12. Se le informa al alumno que allí finaliza el trámite de inscripción.

### **Procedimiento para inscribir a los alumnos irregulares de reingreso**

#### *Propósito*

Contar con una metodología que le permita a la Universidad inscribir a sus alumnos de manera rápida y eficiente.

#### *Procedimiento*

En este procedimiento se involucran las direcciones de servicio escolar, tesorería y se crea un equipo de firmas el cual esta formado por personal administrativo y coordinadores de carrera, el procedimiento se explica con base en el diagrama de flujo 2.4.

#### *A. El equipo de asistentes de firmas: (Punto A)*

1. Se verifica que la matrícula del alumno no aparezca en la “relación de alumnos con adeudos pendientes”. Si aparece su matrícula en ese listado le dirá al alumno que no puede inscribirse en ese momento sino hasta después del periodo de inscripciones, una vez que esté al corriente en sus pagos. Si no aparece la matrícula en el listado puede continuar el procedimiento.
2. Se entrega al alumno los recibos de pago de colegiaturas y le presta un bolígrafo de tinta negra para que éste los firme cada uno.
3. Se verifica que el alumno haya firmado cada uno de los recibos de pago.
4. El personal que atiende al alumno desprende los recibos de pago originales y se queda con ellos.
5. Se entrega al alumno las dos copias restantes y le indica que en ese momento debe pagar el recibo de inscripción, ya que de lo contrario se le dará de baja automáticamente.
6. Se le indica al alumno que se dirija al punto B para continuar el procedimiento.

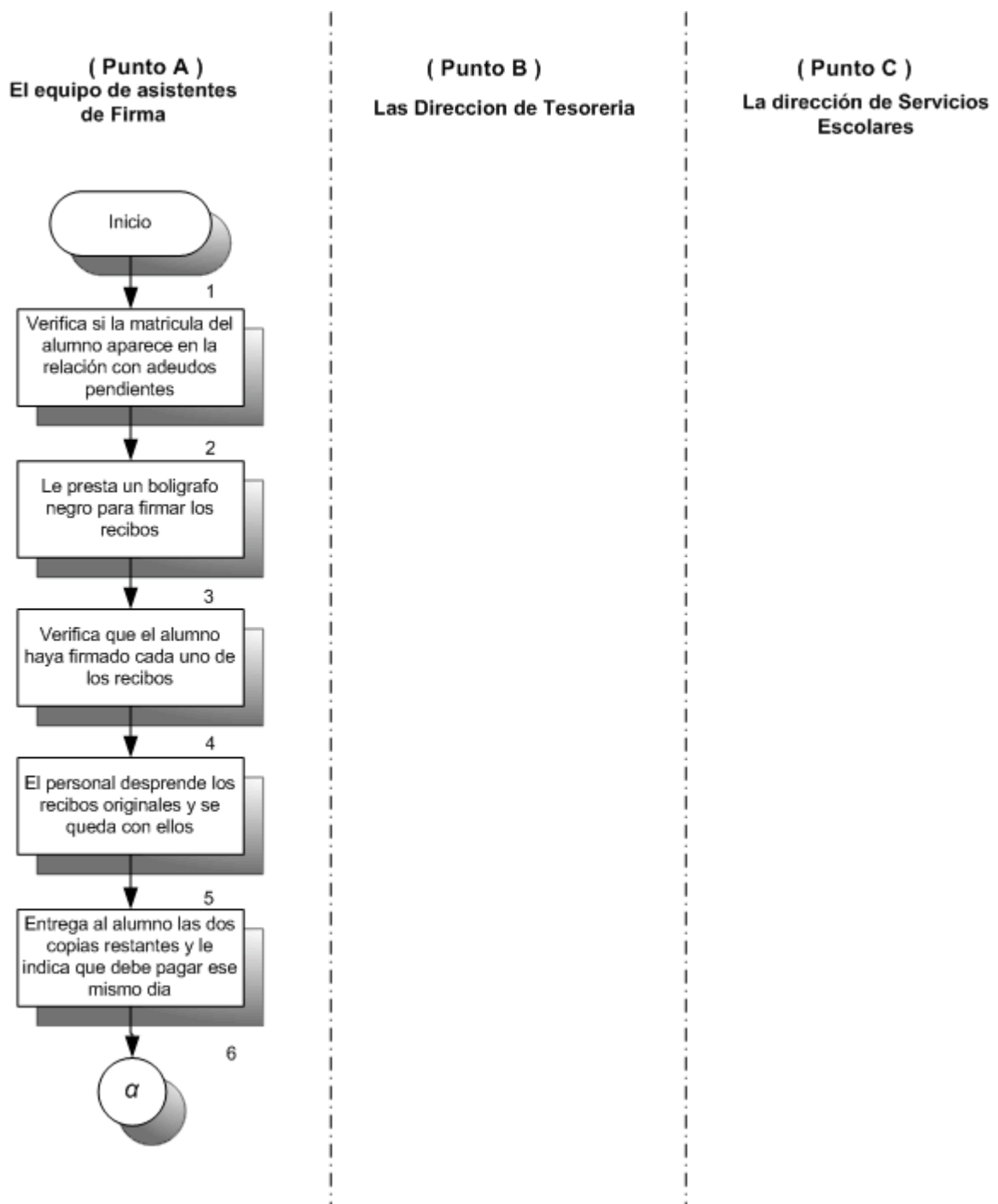


Figura 2.4. Procedimiento de inscripción para alumnos irregulares (1 de 3).

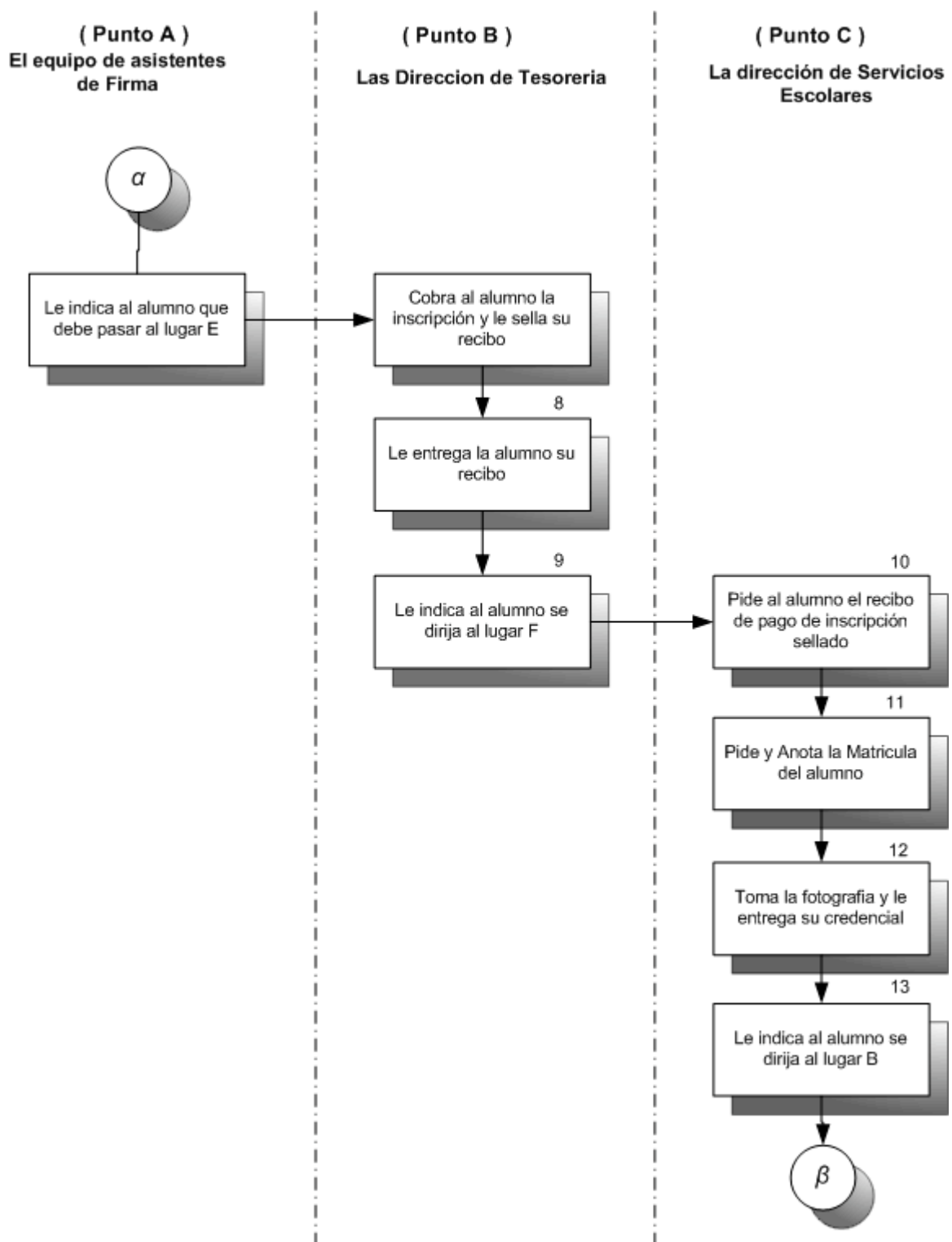


Figura 2.4. Procedimiento de inscripción para alumnos irregulares (2 de 3).

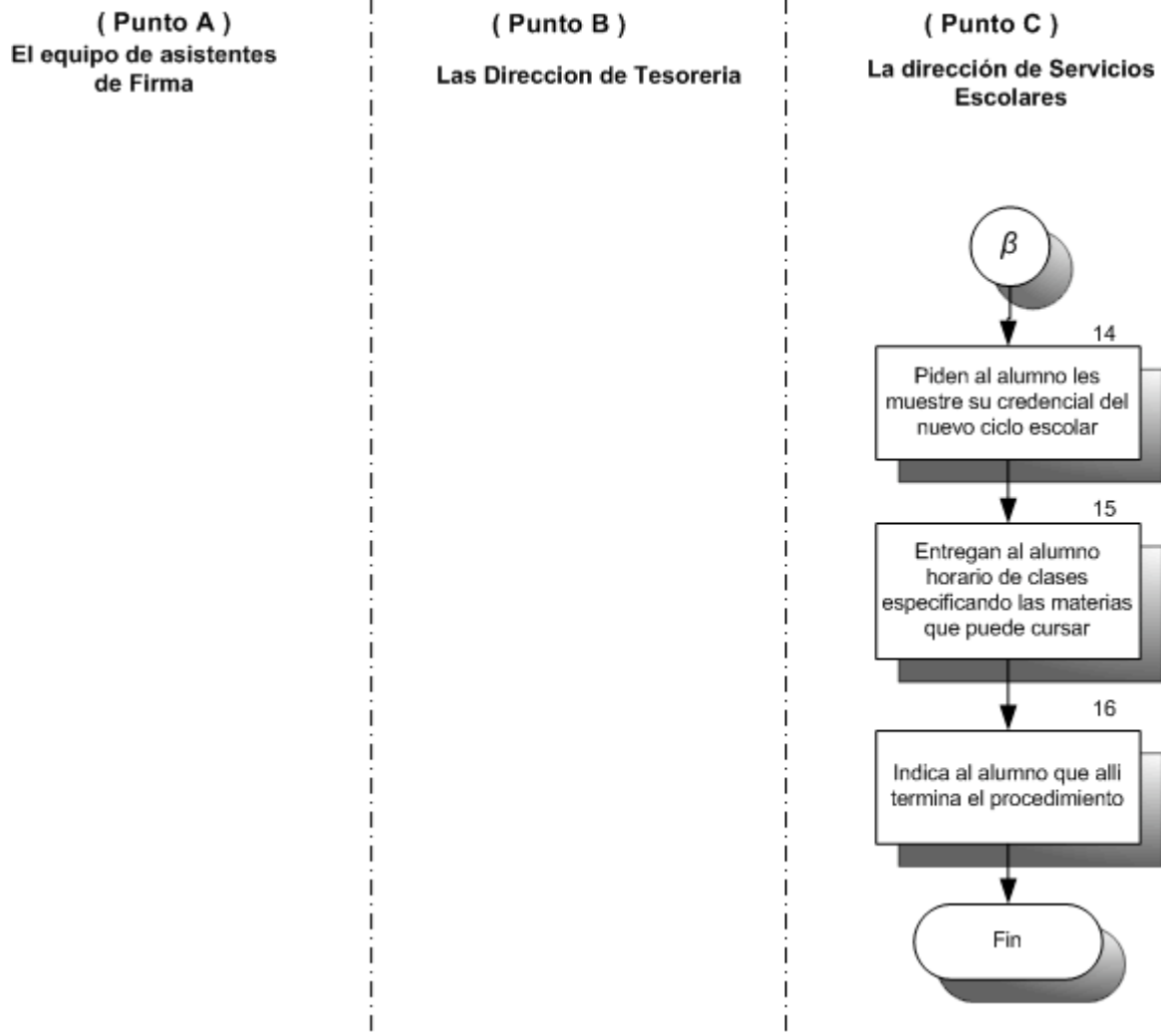


Figura 2.4. Procedimiento de inscripción para alumnos irregulares (3 de 3).

**B. La Dirección de Tesorería: (Punto B)**

7. Se realiza el cobro al alumno regular de reingreso por concepto de inscripción y sella el recibo de pago con la leyenda "PAGADO" y con la fecha del día de pago. En caso de que el alumno solicite una factura se le pide que pase por ella en la primera semana de clases.
8. Se le entrega al alumno el recibo de pago de la inscripción sellado con la leyenda "PAGADO" y la fecha de pago.
9. Se le indica al alumno que se dirija al punto C para continuar el procedimiento.

*C. La Dirección de Servicios Académicos: (punto C)*

10. Se pide al alumno el recibo de pago de la inscripción sellado con la leyenda "PAGADO" y la fecha de pago.
11. Se pide y anota el número de matrícula del alumno para llevar un control interno.
12. Se toma la fotografía y le entrega la credencial al alumno.
13. Se entrega al alumno el horario de clase (señalándole con un plumón verde fluorescente las materias que sí puede cursar), un paquete promocional y un calendario del ciclo escolar.
14. Se le indica al alumno que allí finaliza el procedimiento de inscripción.

**Procedimiento para integrar el expediente físico del personal docente**

*Propósito*

Contar con una metodología que le permita a la Universidad actualizar un expediente físico y electrónico de su planta docente, útil y confiable para las instancias que lo consulten.

*Procedimiento*

El procedimiento para integrar el expediente del personal docente lo realiza la Dirección de Recursos Humanos, apoyada por la Direcciones de Carrera y Área Académica. Este procedimiento se puede observar el diagrama en la figura 2.5.

*A. La Dirección de Recursos Humanos*

1. Se Pide los documentos señalados en registro de candidatos, una vez que el profesor ha sido seleccionado.
2. Se prepara un fólder (expediente físico) con el nombre del profesor para archivar las copias de los siguientes documentos:
  - Acta de nacimiento.
  - Grados académicos.
  - Diplomados.
  - Solicitud de empleo.

3. Se integra la documentación respectiva, conforme el profesor se la entregue.
4. Se le asigna al profesor una clave de identificación única para el control interno del personal docente.
5. Se integra el contrato del profesor a su expediente físico.
6. Se notifica y envía copia del expediente a la dirección correspondiente, de acuerdo al perfil del profesor se envía a la Dirección de Carrera o Área Académica con al menos dos semanas de anticipación al inicio del semestre.

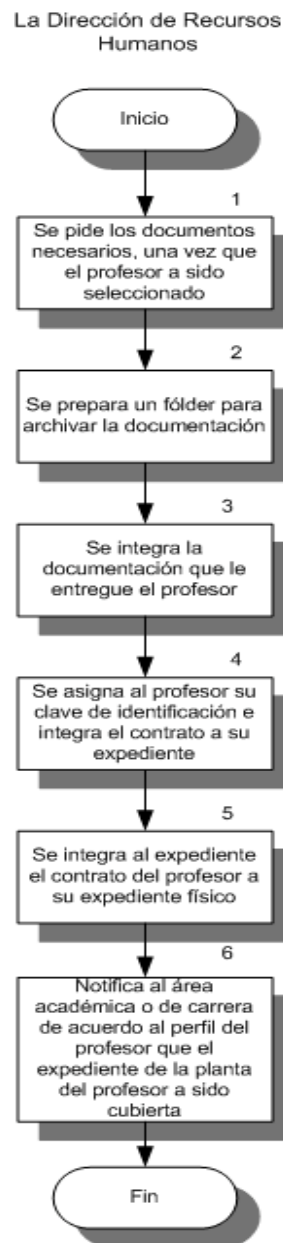


Figura 2.5. Procedimiento para integrar el expediente físico del personal docente.

## **Procedimiento para diseñar y rediseñar planes de estudio**

### *Propósito*

Ofrecer una metodología para regular el diseño y rediseño de planes de estudio.

### *Procedimiento*

En el procedimiento para el diseño y la reestructuración se involucran las Direcciones de Servicio Escolar, Académica de Área, el Consejo Directivo y se crea un equipo de firmas, el cual está integrado por personal administrativo y coordinadores de carrera. El procedimiento se explica con base en el diagrama de flujo 2.6.

#### *A. La Dirección Académica de Área*

1. Se formula en un documento de 5 hojas como máximo, la propuesta para el diseño del plan de estudios, que debe contener:
  - Objetivo.
  - Justificación.
  - Metodología.
  - Cronograma de actividades.
  - Recursos monetarios, físicos y humanos necesarios.
  - Metas y plazos.
2. Se envía al Consejo Directivo el documento creado para su evaluación.

#### *B. El Consejo Directivo*

3. Se evalúa el documento en un plazo máximo de 1 semana y en su caso, lo aprueba. De no ser así, hace las observaciones correspondientes y las envía a la Dirección Académica de Área para que replantee su documento en el paso 1.



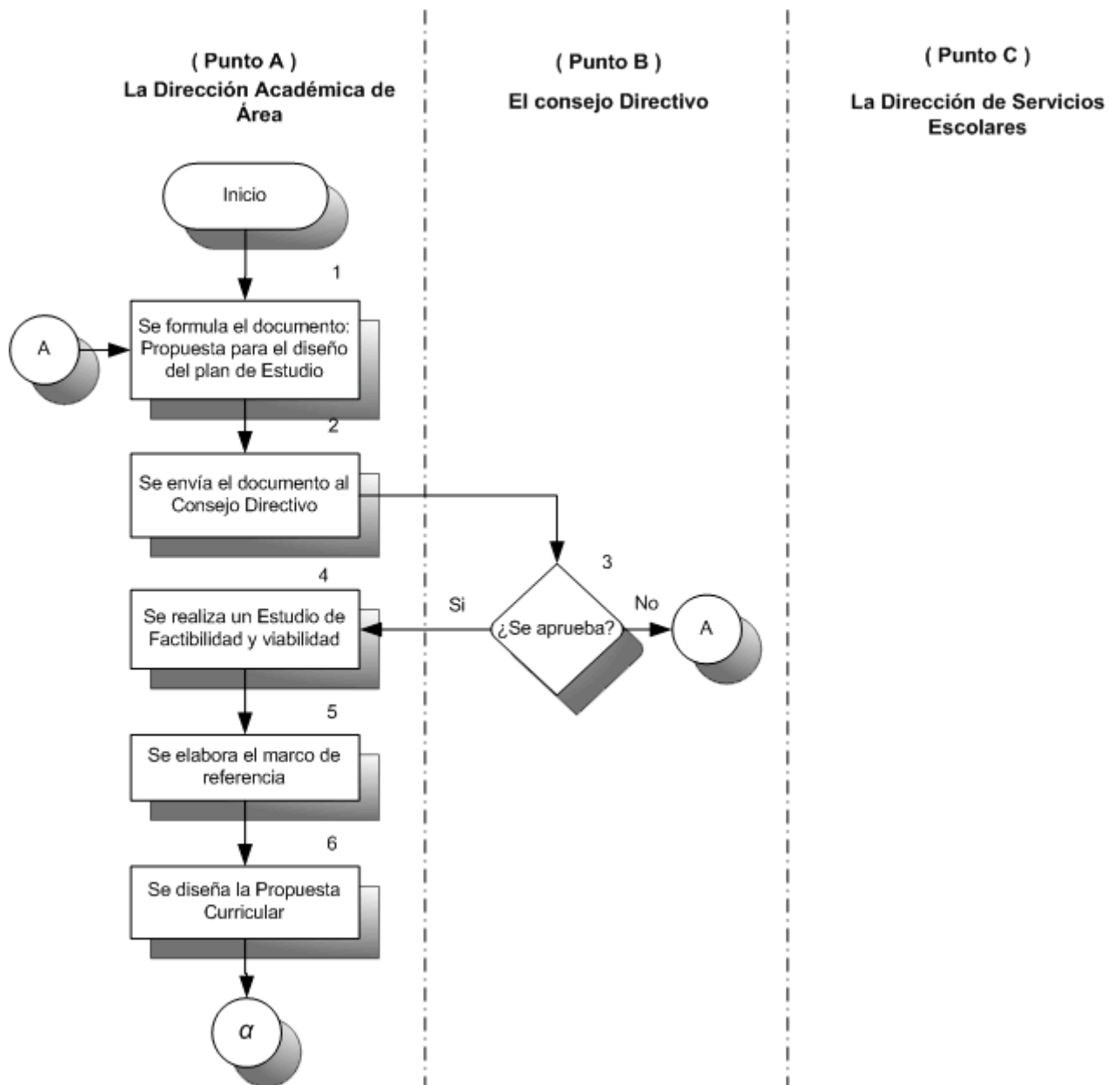


Figura 2.6. Procedimiento para diseñar y rediseñar planes de estudio (1 de 2).

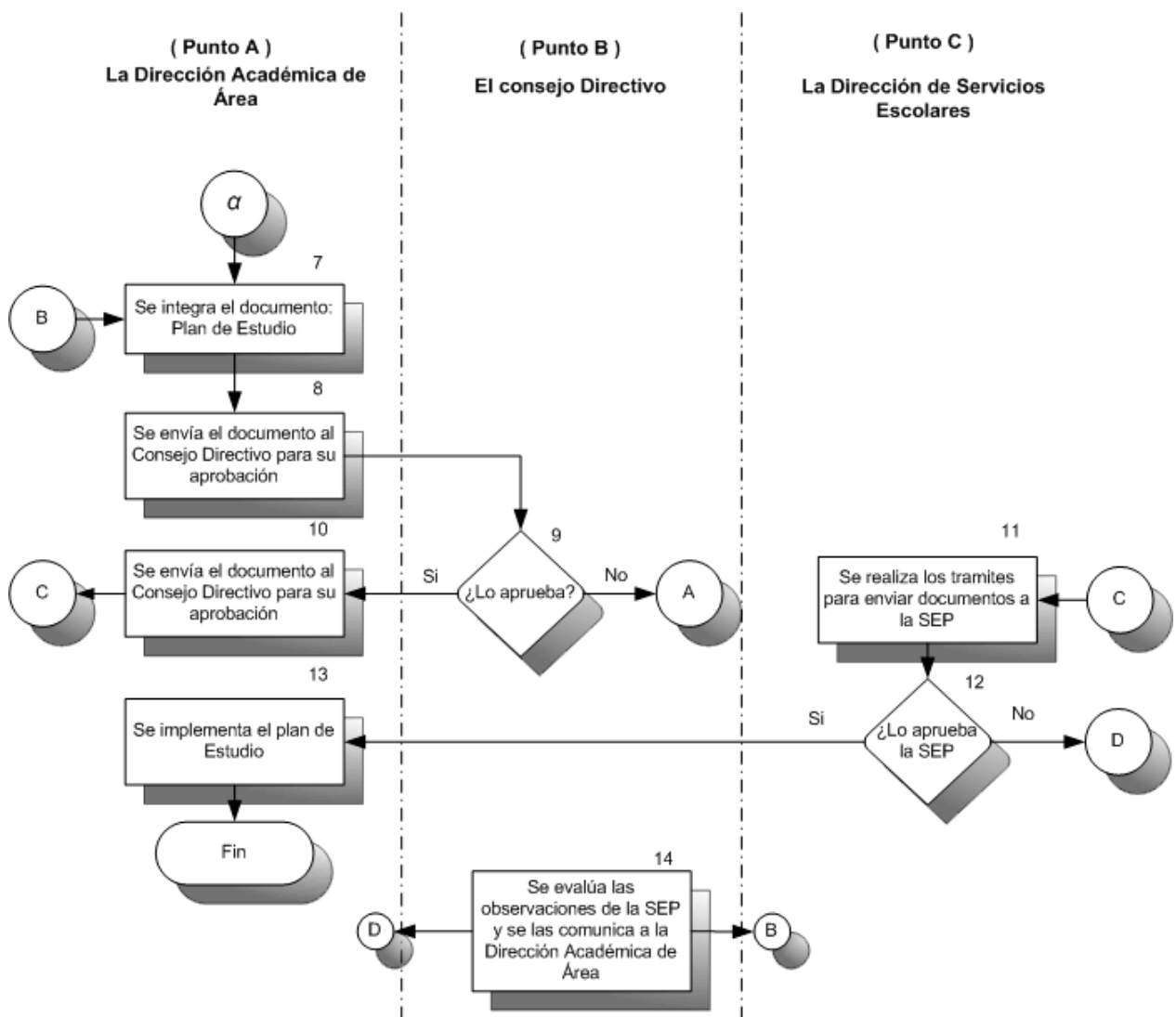


Figura 2.6. Procedimiento para diseñar y rediseñar planes de estudio (2 de 2).

### C. La Dirección Académica de Área:

4. Aprobado el documento, se realiza un estudio de factibilidad y viabilidad para fundamentar el plan de estudio, que debe contener la siguiente información:
  - Análisis socioprofesional.
  - Análisis epistemológico.
  - Análisis psicopedagógico.
  - Análisis de costos.

5. Con base en los resultados anteriores, se elabora el Marco de referencia del plan de estudio, que deberá contener:
  - Justificación de la propuesta curricular.
  - Fundamentación de la propuesta mediante una caracterización del contexto en que se instrumentará el plan de estudio.
6. A partir del marco de referencia se diseña la Propuesta curricular:
  - Formula los objetivos de la carrera.
  - Elabora los perfiles curriculares (de ingreso, de egreso y de docentes)
  - Define los lineamientos curriculares.
  - Selecciona y estructura los contenidos de enseñanza.
  - Define las orientaciones didácticas.
  - Define los criterios de implementación y evaluación del plan de estudio.
7. Con base en la información generada en los pasos 4, 5, y 6 se integra el documento denominado Plan de estudio, que debe contener:
  - Datos generales.
  - Índice.
  - Presentación.
  - Introducción.
  - Metodología desarrollada para la elaboración de la propuesta.
  - Fundamentación de la propuesta.
  - Presentación de la propuesta:
    - Objetivos generales.
    - Perfiles curriculares.
    - Descripción general del plan de estudio.
    - Estructura curricular.
    - Mapa curricular.

- Relación de materias o asignaturas.
  - Programas de materia.
  - Criterios de implementación del plan de estudio.
  - Criterios de evaluación del plan de estudio.
8. Se envía al Consejo Directivo este documento para su evaluación.

*D. El Consejo Directivo*

9. Se evalúa el documento en un plazo máximo de 2 semanas y en su caso, lo aprueba. De no ser así, hace las observaciones correspondientes a la Dirección Académica de Área para que replantee su documento en el paso 7.

*E. La Dirección Académica de Área*

10. Se envía el documento aprobado a la Dirección de Servicios Escolares para que ésta realice los trámites correspondientes con la finalidad de que el plan de estudio sea validado por la Secretaría de Educación Pública (SEP).

*F. La Dirección de Servicios Escolares*

11. Se Realiza los trámites correspondientes para enviar a la SEP el documento Plan de estudio.
12. En el caso de ser validado el documento por la SEP, es enviado a la Dirección Académica de Área para su implementación (13). Si no es validado, debe enviarlo al Consejo Directivo (14).

*G. La Dirección Académica de Área*

13. Se implementa el plan de estudio validado.

*H. El Consejo Directivo*

14. Se evalúa las observaciones hechas por la SEP y regresa el documento a la Dirección Académica de Área para que realice las modificaciones necesarias (paso 7).

**Procedimiento para diseñar y rediseñar los programas de las materias**

### *Propósito*

Ofrecer una metodología para regular el diseño y rediseño de los programas de las materias.

### *Procedimiento*

El procedimiento para diseñar y rediseñar los programas de las materias lo realizan las Direcciones de Servicio Escolar, Académica de Área, y se crea un equipo de profesores, con los conocimientos y experiencia suficiente, para realizar la propuesta y el análisis para crear el programa de materia. El procedimiento se muestra en la figura 2.7.

#### *A. La Dirección Académica de Área*

1. Se define la necesidad de diseñar o rediseñar los programas de las materias.
2. Se integra el equipo de profesores para el diseño o rediseño de los programas de las materias.

#### *B. El equipo de profesores:*

3. Con base en la propuesta curricular del plan de estudio, se define el marco de referencia y se fundamenta el curso, considerando:
  - Los aportes de la materia a la formación profesional.
  - El ciclo, área y eje curriculares en que se ubica la materia.
  - Los cursos antecedentes y consecuentes con que se relaciona.
  - Modalidad.
4. A partir del marco de referencia y la fundamentación del curso, deriva:
  - Las ideas básicas a nivel de aprendizajes deseables.
  - Los propósitos generales de la materia.
  - Los objetivos de aprendizaje como productos.

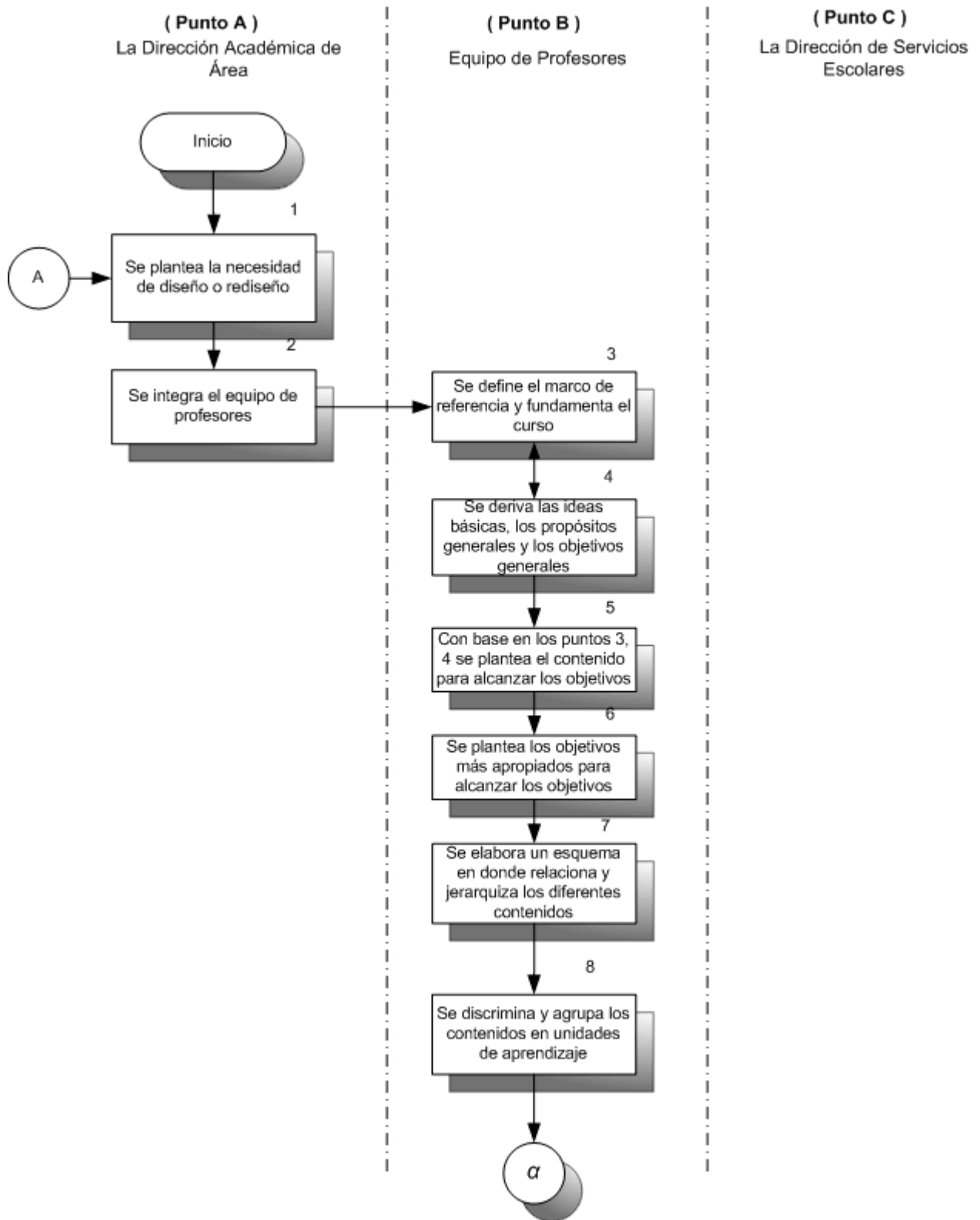


Figura 2.7. Procedimiento para diseñar y rediseñar los programas de las materias (1 de 3).

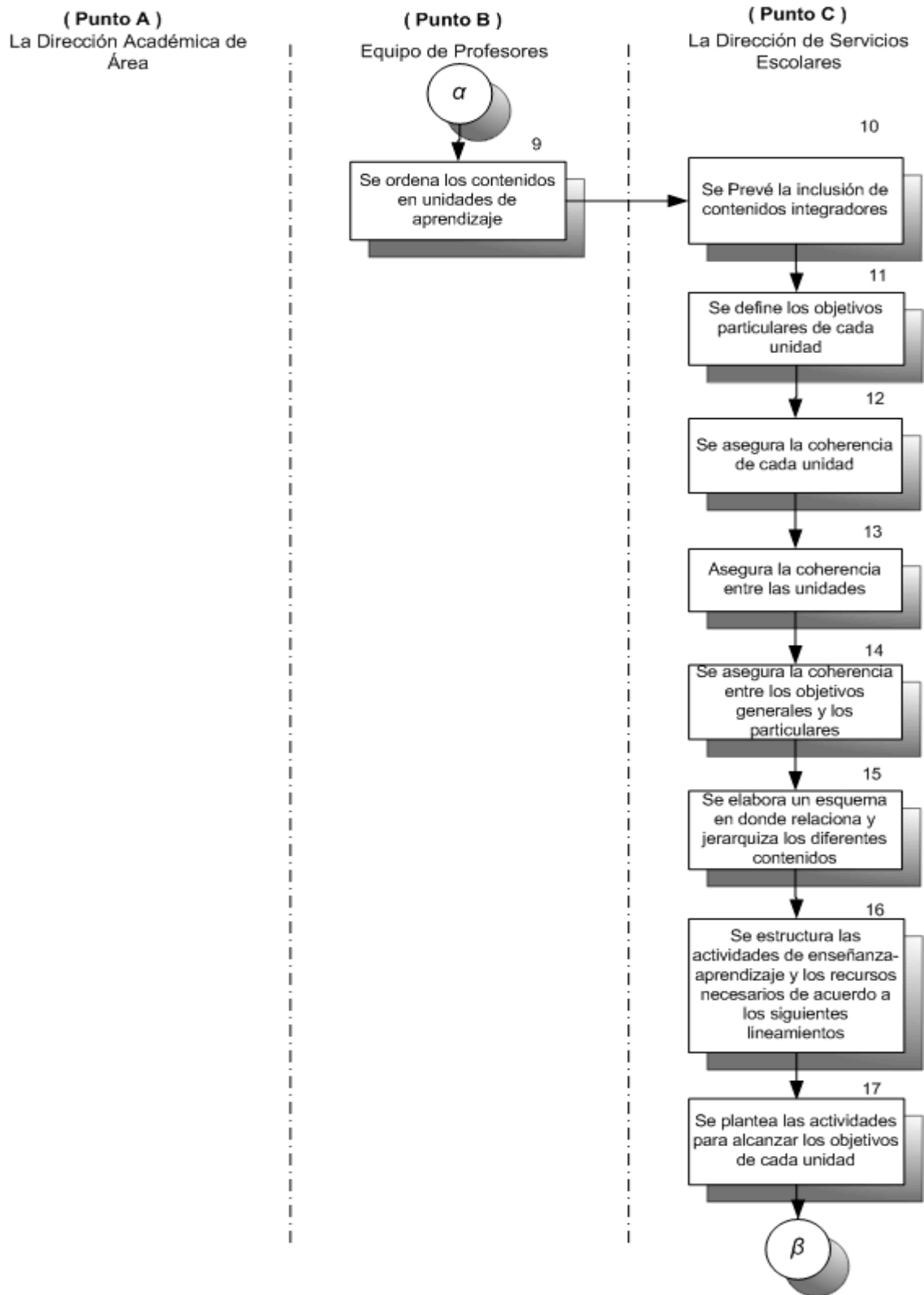
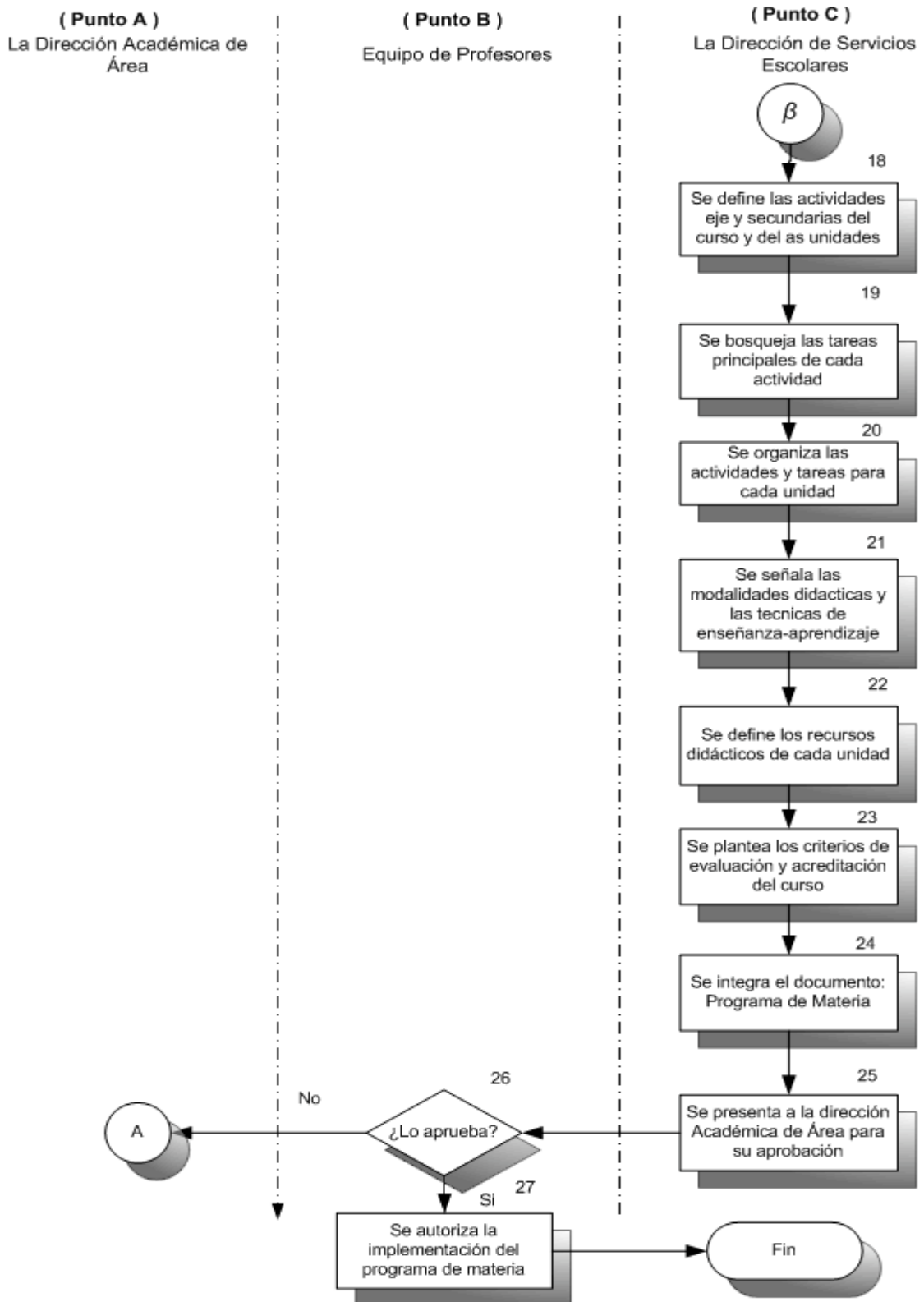


Figura 2.7. Procedimiento para diseñar y rediseñar los programas de las materias (2 de 3).



2.7. Procedimiento para diseñar y rediseñar los programas de las materias (3 de 3).



5. Con base en los puntos 3 y 4 se plantea los contenidos más adecuados para alcanzar los objetivos propuestos de acuerdo a su:
  - Validez: científica, técnica, cultural y práctica.
  - Relevancia: social y profesional.
  - Actualidad.
  - Continuidad con otros cursos del plan de estudio.
  - Grado de generalidad: que incluya dos o más elementos de contenido.
6. Se elabora un esquema o mapa que denote las relaciones existentes entre los diferentes contenidos, así como su jerarquización según el grado de generalidad y abstracción.
7. Se discrimina y agrupa los contenidos en unidades de aprendizaje.
8. Se determina las ideas, temas o aspectos que constituirán los núcleos de cada unidad, con el propósito de que puedan funcionar como organizadores o articuladores de los contenidos particulares.
9. Se ordena los contenidos en unidades de aprendizaje, seleccionando primero aquellos que proporcionen al alumno un panorama, esquema, contexto o punto de partida del curso o unidad; es decir, ideas generales, procedimientos básicos y descripción de situaciones concretas.
10. Se prevé al término de cada unidad y para el curso en su totalidad, la inclusión de contenidos que sinteticen o integren a los desarrollados en la unidad o curso, según corresponda.
11. Se define los objetivos particulares de cada unidad y se expresan las acciones, comportamientos, operaciones intelectuales y competencias que el alumno deberá adquirir o desarrollar con respecto a los contenidos esenciales de cada unidad.
12. Se analiza y jerarquiza los objetivos particulares.
13. Se asegura la coherencia entre los objetivos y contenidos de cada unidad.
14. Se asegura que la vinculación y continuidad de los objetivos y contenidos de las distintas unidades sea coherente.
15. Se asegura la coherencia entre los objetivos generales y los particulares.
16. Con base en el desarrollo de las unidades del curso, se define y estructura las actividades de enseñanza-aprendizaje, así como los recursos bibliográficos,

hemerográficos, electrónicos y didácticos requeridos, de acuerdo con los siguientes lineamientos:

17. Se plantea las actividades necesarias para alcanzar los objetivos generales del curso y los particulares de cada unidad, teniendo en cuenta la modalidad y la naturaleza de los contenidos a desarrollar.
18. Se define, se jerarquiza y estructura las actividades eje y secundarias para el curso y las diversas unidades.
19. Se bosqueja por escrito las tareas principales que se derivarán de las actividades de aprendizaje que planeó.
20. Se organiza las actividades y tareas para cada una de las unidades, de acuerdo a su nivel de complejidad, de inclusión y objetivos de aprendizaje, de manera que plantea las siguientes actividades de:
  - Apertura: para sondear y detectar aprendizajes previos de los alumnos.
  - Desarrollo: para adquirir información nueva y, para desarrollar habilidades y competencias profesionales.
  - Cierre: para aplicar, transferir o generar conocimientos de diverso alcance teórico, técnico y práctico, y para conectar aprendizajes previos con las experiencias generadas en el curso o unidad.
21. Se señala las modalidades didácticas que tendrán las actividades de aprendizaje, como pueden ser: exposición magisterial, asesoría, estudio independiente, entre otros y las técnicas que se utilizarán: tormenta de ideas, debate, estudio de casos, sesión bibliográfica, reportes de lectura, ensayo, investigación, etc.
22. Se define los recursos didácticos que se emplearán en cada una de las unidades, por ejemplo libros, revistas, artículos, películas, audio casetes y direcciones electrónicas.
23. Con base en el análisis de los objetivos generales y particulares de cada unidad, y de las actividades de aprendizaje anteriormente definidas, se plantea los criterios, modalidades y ponderación de la evaluación y acreditación.
24. Con la información generada en los puntos 20, 21, 22, 23 y 24 integra el documento denominado programa de materia, que debe contener:
  - Datos de identificación:
    - Nombre, clave, modalidad y carácter del curso (obligatorio u optativo).

- Clave del plan de estudio.
- Nombre del área académica.
- Nombre del área curricular y semestre en que se ubica el curso.
- En su caso, el valor en créditos.
- El número de horas teóricas y prácticas requeridas para su desarrollo.
- En su caso, los requisitos de seriación.
- Programa sintético:
  - Fundamentación del programa, en el que se ofrecerá una panorámica del curso y se explicarán cuáles son las contribuciones curriculares, asimismo se señalarán los antecedentes y consecuentes de la materia.
  - Objetivos de aprendizaje.
  - Estructura de los contenidos.
  - Presentación de los contenidos en unidades de aprendizaje.
  - Orientaciones metodológicas de los procesos de enseñanza-aprendizaje.
  - Modalidades, criterios y requisitos de evaluación y acreditación.
  - Recursos didácticos y bibliografía requeridos.
  - Perfil académico deseable del profesor que impartirá el curso.
- Datos de cierre:
  - Fecha del diseño o rediseño del programa.
  - Nombre del diseñador o rediseñador.

25. Presenta el documento a la dirección académica para su aprobación.

### *C. La Dirección Académica de Área*

26. Evalúa el documento en un plazo máximo de 1 semana y en su caso, lo aprueba. Si no lo aprueba le hace las observaciones correspondientes al equipo de profesores para que replantee su documento en el paso 1.

27. Autoriza la implementación del “Programa de Materia”.

La mayoría de la información se encuentra clasificada en archiveros mediante el manejo de expedientes, dentro de los cuales podemos encontrar documentación personal y de control como:

- Profesores. De estos se mantiene información en base a expedientes que contienen los siguientes documentos:
  - Acta de nacimiento.
  - Cartilla militar.
  - Hoja de registro de datos.
  - Examen sicométrico.
  - Hoja de control de asistencia.
  - Historial de vida curricular.
  - Cedula profesional.
  - Comprobante de domicilio.
  - Examen médico.
  - Hoja rosa del IMSS.
  - Registro federal de causantes.
- Alumnos. Los expedientes de los alumnos se clasifican de acuerdo a su año de ingreso, carrera y nombre, en estos encontramos los siguientes documentos:
  - Acta de nacimiento.
  - Último comprobante de estudios.
  - Comprobante de domicilio.
  - Hoja de datos del padre o tutor.
  - Examen médico.
  - Boleta de pagos.
  - Hoja de registro con datos personales.

- Hoja de inscripción.
- Historial de su vida académica.
- Planes de estudio. Se guarda la descripción, objetivo, justificación y mapa de asignaturas o materias que comprende a cada plan.
- Materias. Se guarda el temario por materia, organizados por plan de estudios.
- Periodo Escolar. En éste se guarda la siguiente información:
  - Grupos. Por cada grupo se guarda la lista de alumnos y la lista de asistencia por materia.
  - La lista de asignación de profesores por materia y horario ordenado por carrera y plan de estudios.

## **2.2. Definición de la problemática**

En la actualidad la Institución Educativa presenta serias dificultades al inicio de los ciclos escolares, ya que no cuenta con un sistema que le permita la manipulación de información para un mayor control de actividades, como son la actualización, consulta y búsqueda de toda información relacionada con:

- Planes de estudio.
- Alumnos.
- Profesores.
- Pagos.

En la institución se generan altos costos al inicio y en el transcurso del ciclo escolar, debido a que se invierten demasiados recursos en tiempo y material para la administración de la información. En el caso de los profesores no se lleva un seguimiento de las incidencias en los que estos incurren, tales como: faltas, retardos, evaluación de los alumnos, entre otras observaciones.

En el caso de los alumnos la creación de comprobantes, constancias, cartas, entre otros documentos, se realizan bajo trámites complejos que pueden llevar varios días para ser atendidos.

Estas cuestiones son algo problemáticas pues, generan retrasos en la fluidez de la información, lo cual causa un servicio poco eficiente para la comunidad estudiantil.

Otro de los problemas es que no existe un control adecuado sobre las plazas que ocupan los profesores. Se requiere que los profesores cumplan con cierto perfil académico de acuerdo a su experiencia docente y humana, para que puedan ser asignados a las distintas áreas del conocimiento, de las diferentes carreras, de la forma más óptima posible.

Con base en lo anterior, es evidente la necesidad de contar con un sistema que tenga como objeto el administrar la información y permitir la optimización de los recursos, tanto materiales como humanos, con los que se cuenta actualmente.

En la actualidad la institución emplea gran cantidad de papel para el manejo de información, es decir, se tienen listas y documentos de profesores, alumnos, grupos, materias, horarios, grados académicos y planes de estudio. Es evidente la necesidad de la existencia de dichos documentos en búsqueda de las posibles aclaraciones o trámites en los que se encuentren involucrados los alumnos, sin embargo en la operación esto es poco práctico, debido a que el manejo de tal volumen de información es muy complejo y poco funcional. Este problema se ha presentado por el crecimiento en la plantilla de alumnos ocasionada por el paso del tiempo.

La hoy necesaria actualización de los planes de estudio, la mejora continua de los mismos, el cambio de profesores, el manejo de la información de los alumnos y la reciente implantación de una de una red de computadoras en el área administrativa, son la causa principal para desarrollar un sistema administrador de información escolar.

Considerando sus principales actividades y la forma en que las llevan a cabo, se identifican los siguientes problemas respecto al manejo de información:

- La captura de la información es laboriosa e inadecuada.
- La búsqueda de información no es ágil aún cuando se tiene clasificada.
- El manejo de documentos en papel hace susceptible que estos se desgasten, por lo cual se pierde legibilidad en ellos.
- Los documentos en papel tienen el riesgo de perderse.
- No se tienen respaldos fiables de información (la escuela sólo maneja un número limitado de copias de las listas de profesores, alumnos, materias, grupos y horarios).
- La asignación de grupos y horarios en un proceso lento y laborioso.
- El costo que implica realizar los procesos de administración de esta información, principalmente al inicio y final de cada ciclo escolar es muy elevado, debido al alto consumo de recursos y al planteamiento de procesos de administración.

- Existe duplicidad de información, debido a que la plantilla de profesores no cambia en gran cantidad en cada ciclo escolar.
- Existe duplicidad de información, debido a que los datos generales de los alumnos no cambian demasiado durante su estancia en la institución.

Este esquema de manejo de información es anticuado y obsoleto, ya que no permite de manera fácil y eficiente la corrección de información y la realización de consultas con la cual se puedan tomar decisiones.

Con base en los puntos mencionados anteriormente, identificamos los siguientes procesos a optimizarse:

- La inscripción, reinscripción y baja de alumnos.
- La administración de la información de datos escolares y personales de alumnos.
- La información referente a los profesores.
- Organización de planes de estudio.
- Materias se realizará con mayor rapidez y confiabilidad en el manejo de la información.

Mediante la optimización de procesos se pretende reducir costos, ahorrar tiempo y aprovechar al máximo los recursos tanto técnicos como humanos con los que cuenta la institución.

### **2.3. Propuesta de Solución y definición del alcance del Proyecto**

En base a los requerimientos especificados en el apartado anterior, definimos la necesidad de desarrollar un sistema de información capaz de cumplir con los siguientes objetivos:

- Diseñar un modelo de datos que contenga la información necesaria de las áreas administrativas de la institución y que permita el correcto funcionamiento de los procedimientos anteriormente mencionados.
- Crear una base de datos en la cual pueda almacenarse la información generada, que cuente con reglas de integridad referencial y esté normalizada.

- Desarrollar un sistema que permita interactuar a los usuarios con el modelo de datos, a fin de realizar modificaciones en cuanto al almacenamiento de la información.
- Crear un esquema de seguridad que permita la modificación de la información basado en los permisos y alcances de cada área.
- Crear un módulo de alumnos que permita agregar, modificar, dar de baja y consultar información sobre los datos de los alumnos. El módulo de alumnos se puede dividir en los siguientes apartados:
  - Datos personales. Aquí se definen los datos para contactar al alumno y sus padres, información relativa a su domicilio particular, teléfonos de su casa y correo electrónico
  - Identificador de alumno. Asignación de un identificador por el cual el alumno sea relacionado de manera única dentro del sistema.
  - Datos de Inscripción. De acuerdo al tipo de alumno, si se trata de un alumno que apenas ingresará a la institución o de un alumno con trayectoria curricular.
- Crear un módulo de profesores que permita agregar, modificar, dar de baja y consultar la siguiente información:
  - Datos Personales: Se tendrá el registro completo sobre el profesor entorno a nombre, edad, sexo, nacionalidad, domicilio, teléfono, entre otros.
  - Datos Clínicos. De acuerdo a políticas de la institución, se pedirá como requisito a profesores aportar cuestiones como el uso de lentes, tipo de sangre, mención de enfermedades padecidas, etc.
  - Descripción del Perfil. Carreras que cursó, grado actual del profesor, especificación de especialidades con base en su experiencia en la docencia.
  - Control. Comportamiento dentro de la institución, asistencias e incidencias.

Adicionalmente, los objetivos generales del sistema de administración, se centran en conseguir una mejor operatividad y consistencia en la información, basados en los siguientes puntos:

- Ser de fácil uso y manipulación.
- Tener una interfaz agradable para el usuario final.
- Realizar el alta, baja y la actualización de información en cada módulo.
- Permitir la consulta detallada y jerarquizada de la información en cada módulo.



- Ser seguro en cuanto al acceso para tener un control de usuarios.
- Evitar la duplicidad de información.
- Evitar la eliminación de la información relacionada.
- Agilizar drásticamente los procesos de búsqueda de información.

Analizando los puntos anteriores podemos evaluar que las interfaces de usuario, la operatividad con la base de datos y los elementos propios de cada módulo individual de la aplicación, son muy parecidas entre sí, por lo cual planteamos el desarrollo de una herramienta generadora de código, que nos permita la creación de la aplicación deseada de forma automática.

Finalmente, una vez que hemos revisado de forma general los requisitos, procedimientos y problemática de la Institución Educativa, es necesario comenzar con la siguiente fase de diseño del sistema de información, que nos permita cubrir los objetivos planteados.

# 3

## Diseño del Sistema

---

La etapa de diseño comprende la definición del modelado del sistema, así como la estructura de los datos y la forma en que éstos se relacionan. A partir de la descripción de las funciones y de la especificación de las interfaces es posible implementar cada parte que compone el sistema. En este capítulo, de acuerdo a los requisitos del sistema, crearemos la estructura de datos y la representación del mismo; especificando la descripción de los datos y definiendo el comportamiento de la aplicación.

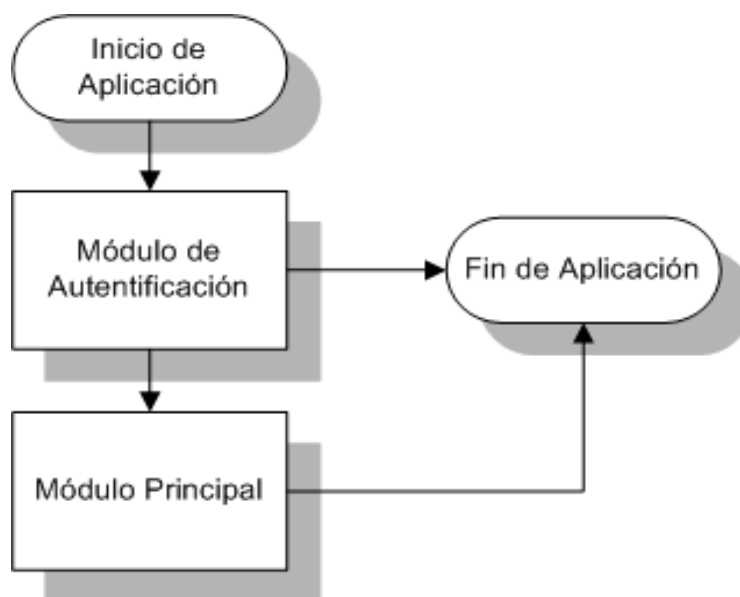
### 3.1. Modelado del Sistema

El modelado lleva una especificación de los requisitos y una representación del diseño general de la aplicación. En el modelado se representa al sistema con diagramas de flujo de datos, se explica a grandes rasgos el funcionamiento del mismo y los procesos que lo conforman.

El sistema, como lo describimos en el capítulo anterior, tiene como propósito administrar la base de información que contiene los datos referentes a alumnos, profesores, plan de estudios y pagos, con la que se realizan actividades y procedimientos importantes para la Institución Educativa.

En la figura 3.1 se muestra el diagrama de flujo del sistema, se observa que éste debe contener una pantalla o interfaz de inicio como presentación del sistema, un módulo de sesión que autentifique el ingreso del usuario para el acceso al módulo principal y que a su vez configure el sistema de acuerdo al perfil del área a la cual pertenece dicho usuario.

El módulo principal tendrá como propósito agrupar los módulos para la manipulación de la información contenida en la base de datos, permitiendo acciones como consulta, borrado, actualización y alta de registros.



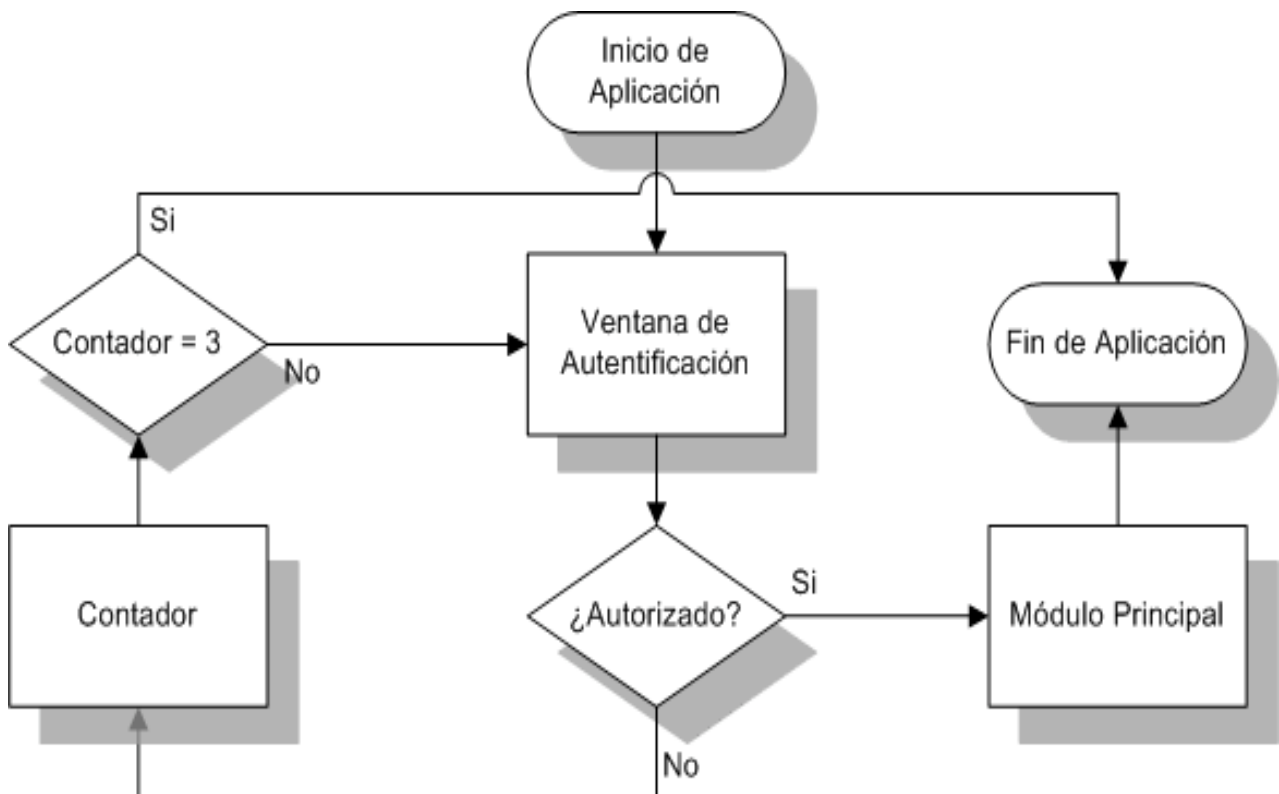
*Figura 3.1. Diagrama General del Sistema.*

Cada uno de los módulos posee las herramientas necesarias para la interacción con la base de datos, aplicando reglas de integridad y validación de información, así como también funciones que permiten realizar consultas complejas con base en un número diverso de criterios y filtros.

En el sistema se definen distintos tipos de usuarios agrupados según la dirección a la que pertenecen, las tareas que realizan y el rol que desempeñan en la escuela, que definen el perfil de permisos en el sistema.

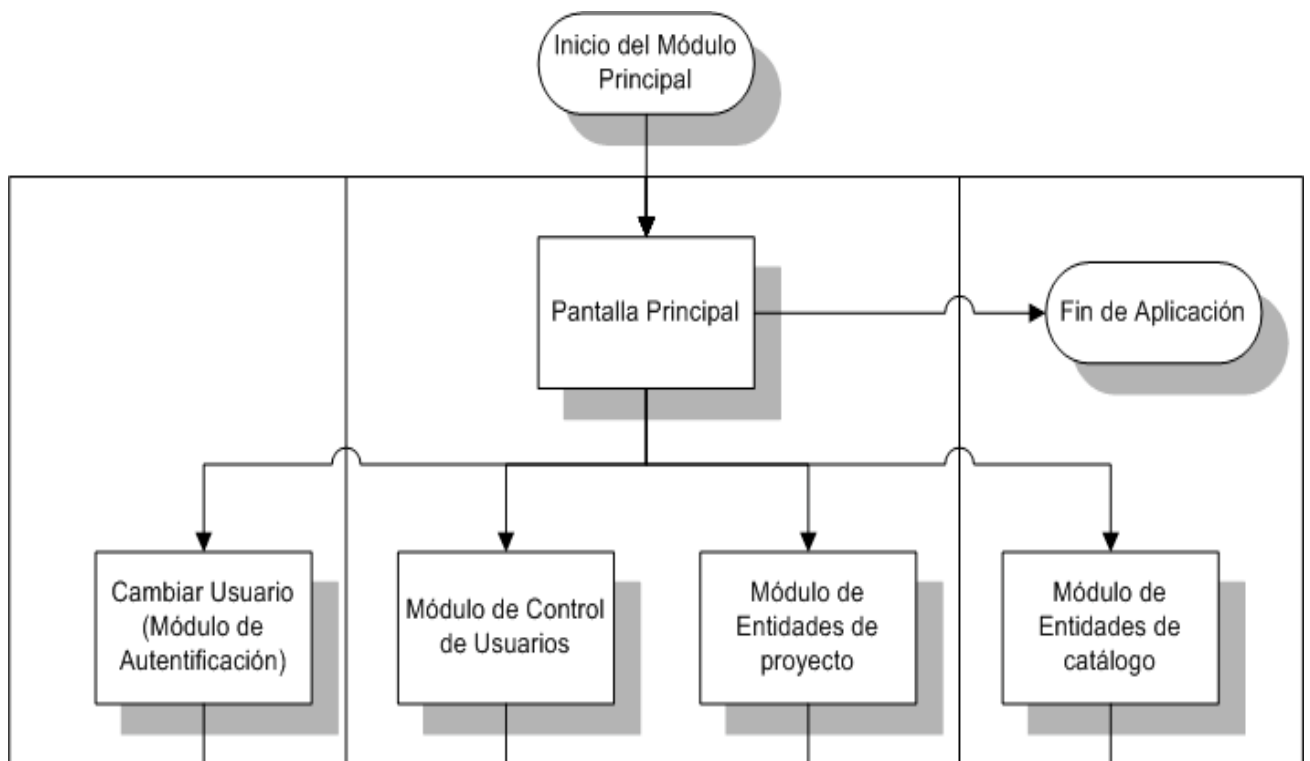
El módulo de inicio de sesión, ver figura 3.2., solicita la autenticación del usuario, pidiéndole teclear su identificador y contraseña para que sean validados por el sistema. En caso de que sean correctos, le es permitido el acceso al módulo o interfaz principal.

Cabe comentar que existe un grupo administrador que tiene todos los privilegios dentro del sistema, es el único que cuenta con usuarios capaces de dar de alta, borrar y modificar los privilegios de otros usuarios.



*Figura 3.2. Módulo de Autenticación de Usuarios.*

Por otra parte, el módulo principal está compuesto por cuatro módulos: módulo de autenticación, módulo de control de usuarios, entidades de proyecto y entidades de catálogos, como puede verse en la figura 3.3.



*Figura 3.3. Diagrama del Módulo Principal.*

#### *Módulo de autenticación*

El módulo de autenticación permite cambiar de usuario sin salir del sistema. Lo cual cambia los permisos y privilegios del usuario al trabajar con los demás módulos.

#### *Módulo de control de usuarios*

Este módulo administra la información de usuarios. Permite la realización de altas, bajas, modificación y consulta de usuarios. Se definen los permisos y privilegios que gozarán cada usuario dentro del sistema administrador.

#### *Módulo de entidades de proyecto / catálogos*

El sistema mostrará un listado de las distintas entidades del proyecto que se desea sean administradas por el sistema: alumnos, profesores, plan de estudios, materias, pagos, etc. Además de mantener los catálogos del sistema, para poder realizar altas, bajas, modificaciones y consultas sobre la información contenida, como: motivos de inasistencia, bancos, etc.

Permitirá, de acuerdo a los privilegios del usuario, visualizar la lista de módulos que le corresponde de acuerdo a las actividades que realice, además dentro de cada

módulo al cual tenga acceso, contará con los mismos o distintos permisos según se haya definido dentro del módulo de usuarios.

Los módulos de entidades de proyecto y de catálogos serán un estándar dentro de la aplicación, lo cual permitirá que el sistema sea homogéneo dentro de sus interfaces, sencillo de manipular y mantendrá un orden dentro de las acciones que realizará sobre la estructura de datos.

Estos módulos se encargan de la administración de la fuente de datos del sistema. La diferencia entre entidades de proyecto y catálogos, radica en la relevancia e impacto de la información dentro del sistema.

En la figura 3.4 se muestra el diagrama de flujo del modelo genérico de los módulos de entidades de proyecto y de catálogos. Al acceder a un módulo específico, ya sea del proyecto o de catálogo, se mostrará el módulo de consulta.

El módulo de consulta muestra los resultados del criterio definido en la entidad a la que se hace referencia en el módulo, es posible modificar la información de los registros contenidos en la consulta seleccionando el registro deseado, lo cual desplegará el módulo de modificación. Este módulo tiene un menú principal con las opciones a las que un usuario en específico tiene permiso, estas opciones pueden ser en su totalidad las siguientes:

- **Buscar.** Esta acción abrirá una nueva interfaz, en la cual se mostrarán los campos por los cuales será posible realizar una búsqueda, por uno o varios campos según lo determine el propio usuario. El resultado de la búsqueda quedará reflejado en la pantalla de consulta.
- **Filtrar.** Esta acción abrirá una nueva pantalla, la cual permitirá de acuerdo a la regla que definamos el criterio de búsqueda. El resultado al aplicar el filtro se visualizará en la pantalla de consulta.
- **Imprimir.** Abre una nueva ventana con la vista previa para impresión de la consulta que se muestra en ese mismo instante en la pantalla de consulta, tiene las opciones de cerrar y enviar a la cola de impresión.
- **Agrupar.** Permite dentro de la pantalla de consulta agrupar las columnas de datos de acuerdo a los criterios especificados por el usuario, para permitir una mayor legibilidad y orden de la información.
- **Columnas.** Realiza un filtrado de columnas, permite mover, ocultar y visualizar las columnas de acuerdo a los intereses que el usuario elija.

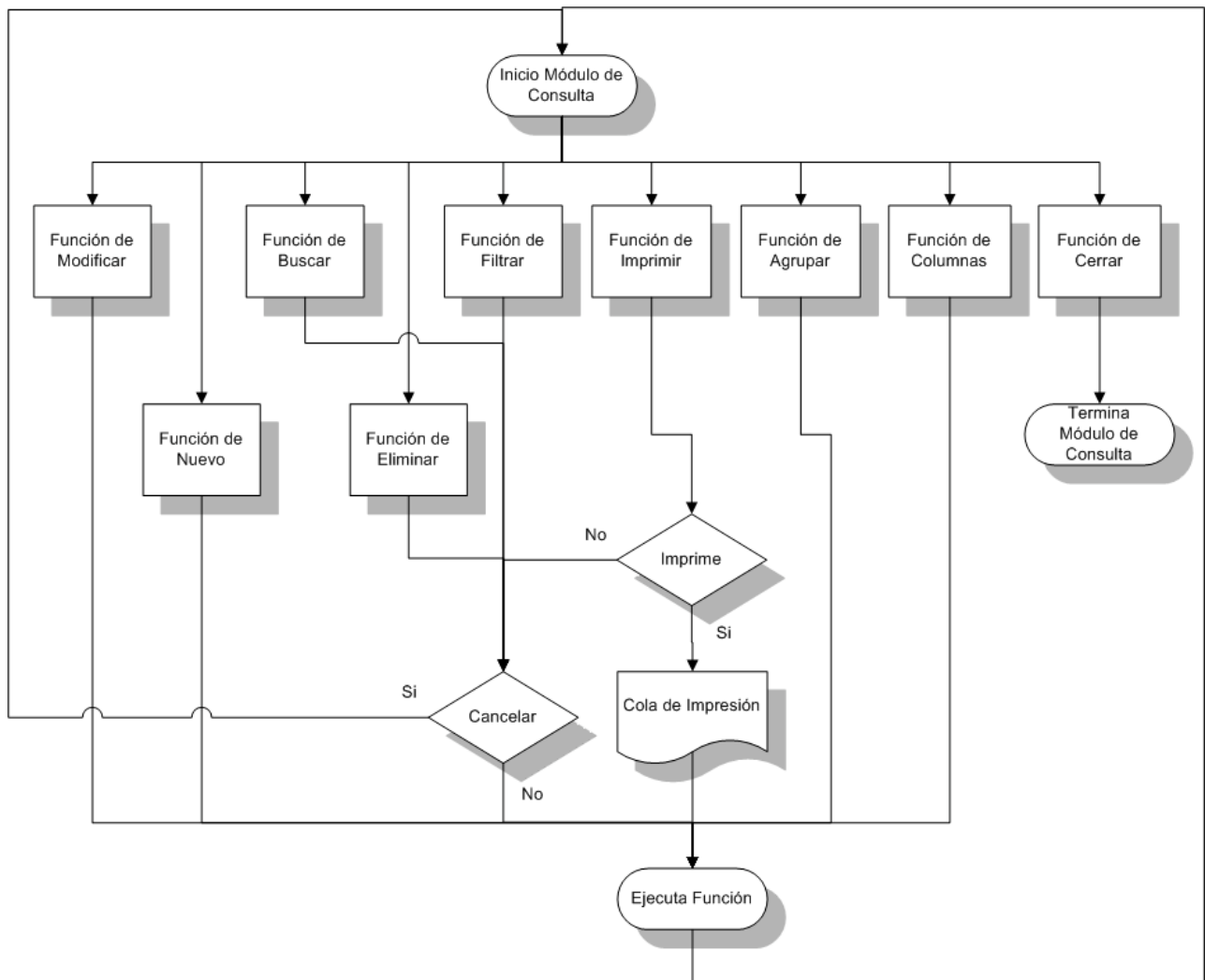


Figura 3.4. Funcionalidad del Módulo de Consulta.

- Eliminar. Permite eliminar los registros seleccionados dentro de la pantalla de consulta.
- Nuevo. Abre una pantalla con los campos necesarios para dar de alta un nuevo registro dentro del sistema, el menú de esta pantalla tendrá opciones como:
  - Cerrar. Cierra la pantalla.
  - Guardar. Guarda los cambios del registro, realiza el alta dentro del sistema y cierra la pantalla “Nuevo”.

- Guarda y Nuevo. Similar al anterior sólo que deja abierta la pantalla y limpia los campos de captura para realizar el alta de un nuevo registro.
- Cerrar. Cierra la pantalla de Consulta.
- Modificar. Permite, dentro de la pantalla, modificar los datos si sus permisos son validados.

Adicionalmente la tabla en la que se desglosan los datos permite el ordenamiento de la información contenida en ella, basándose en la selección de una columna y ordenando la información de la columna de forma ascendente o descendente.

## 3.2. Estructura de Datos

La estructura de datos es la representación lógica entre elementos individuales de datos y sus relaciones. Esto se conoce también con el nombre de modelado de base de datos.

La presentación del modelado de datos en el diseño se divide en los siguientes módulos:

- Autenticación
- Alumnos
- Alumnos-Padres
- Alumnos-Pagos
- Profesores
- Planes de Estudio

### Autenticación

En este módulo se realizará la definición de los grupos y usuarios que accederán al sistema, para cada uno de los usuarios se establecerán los permisos o privilegios sobre las interfaces y sus funciones de consulta, alta, baja y modificación. La información se almacenara de acuerdo a la figura 3.5 que muestra el diagrama E-R de la autenticación.



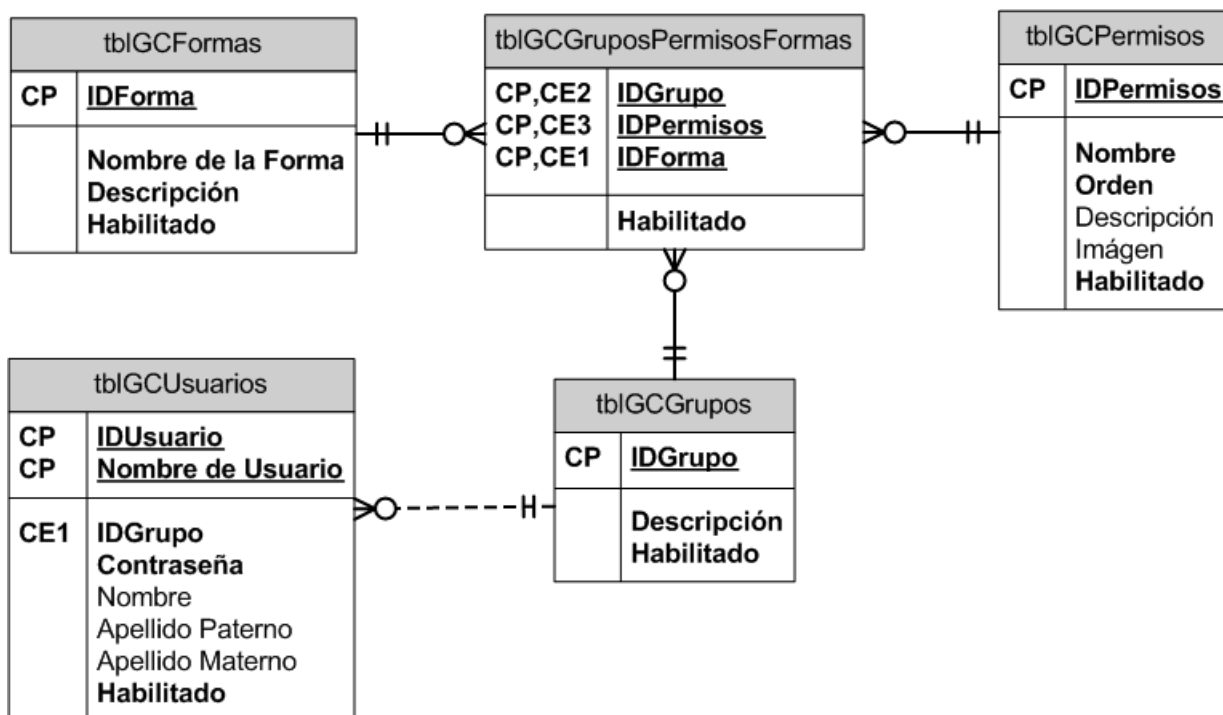


Figura 3.5. Modelo Entidad-Relación para la Autenticación de Usuarios.

En este módulo se almacenará la información del usuario, grupo al que pertenece, descripción de las pantallas o formas a las que puede tener acceso y la definición de permisos para realizar consultas, altas y modificación en las pantallas o formas con que contará el sistema.

## Alumnos

El módulo de alumnos, ver figura 3.6., comprende distintas entidades de las cuales explicaremos las más importantes a continuación:

- Alumnos. En está se incluyen los datos personales del alumno, como son: número de cuenta o matricula, carrera a la cual está inscrito, escuela de procedencia, fotografía, entre otros.
- Medicina General. Incluye un catálogo de enfermedades, tipo de sangre e información clínica que se relaciona con el alumno.
- Documentos entregados. Se refiere a los documentos que el alumno halla entregado a las direcciones de la escuela indicando la fecha, la cantidad y el tipo de documento.

- Documentos Recibidos. Se refiere a los documentos que el alumno halla recibido de la escuela indicando la fecha, la cantidad y el tipo de documento.
- Status. Se refiere a las observaciones generales que el alumno puede presentar.
- Mail. Se refiere a las diferentes direcciones electrónicas que tiene un alumno.
- Carreras. Contiene la descripción de las distintas carreras y la relación con el alumno, indica la carrera que le corresponde a cada alumno.
- Exámenes. Se refiere a los exámenes realizados por el alumno, indicando la fecha de realización, la calificación y el tipo de examen.
- Teléfonos. Se refiere a los distintos números telefónicos y las características especiales de estos, para localizar al alumno.

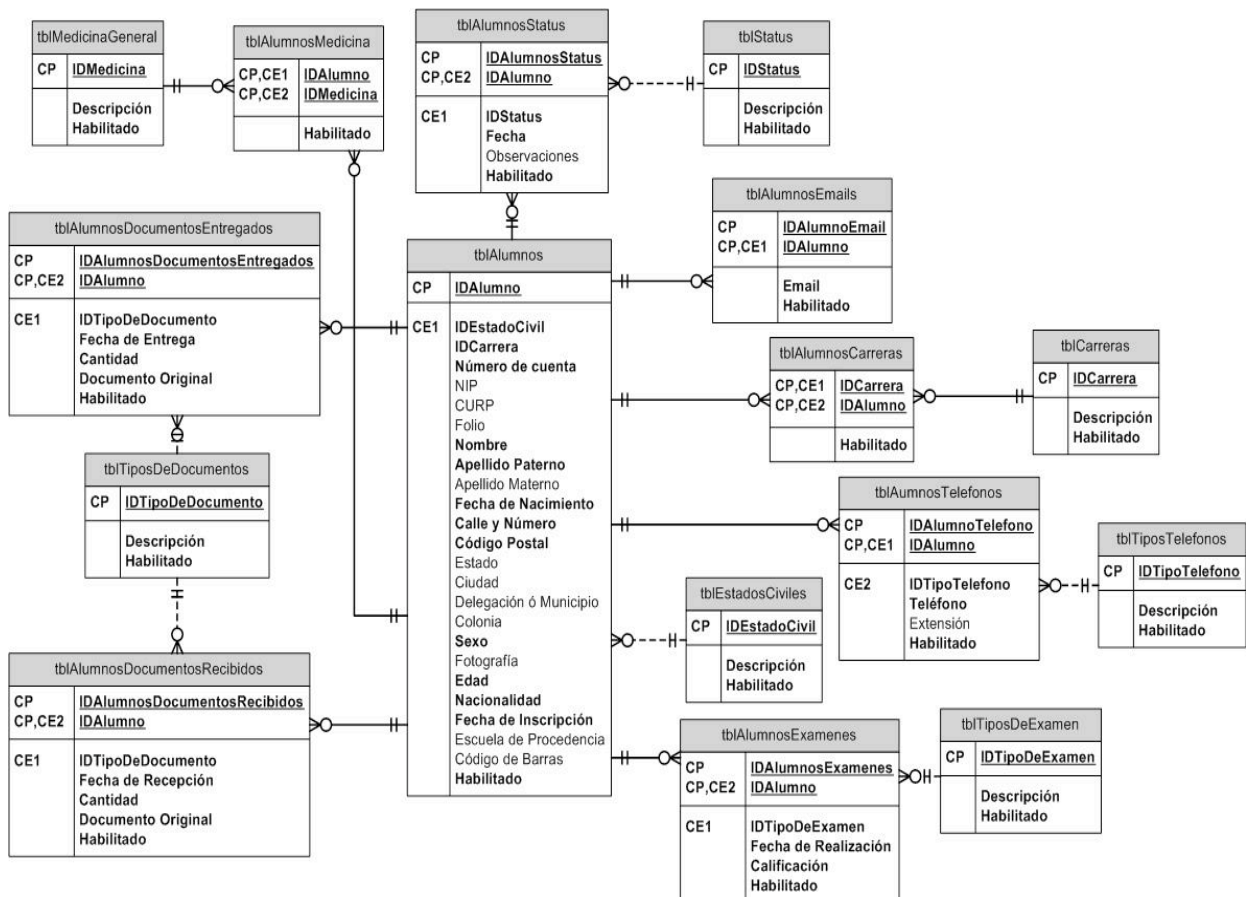


Figura 3.6. Modelo E-R para el Módulo de Alumnos.

## Alumnos Padres

En este módulo presentamos las entidades, ver figura 3.7., que relacionan al padre o tutor con el alumno. Las entidades principales del padre o tutor del alumno se describen a continuación:

- **Padres.** Se refiere a los datos personales, datos labores, profesión, domicilio, entre otros.
- **Teléfonos.** Se refiere a los distintos números telefónicos y las características especiales de estos, para localizar al padre o tutor.

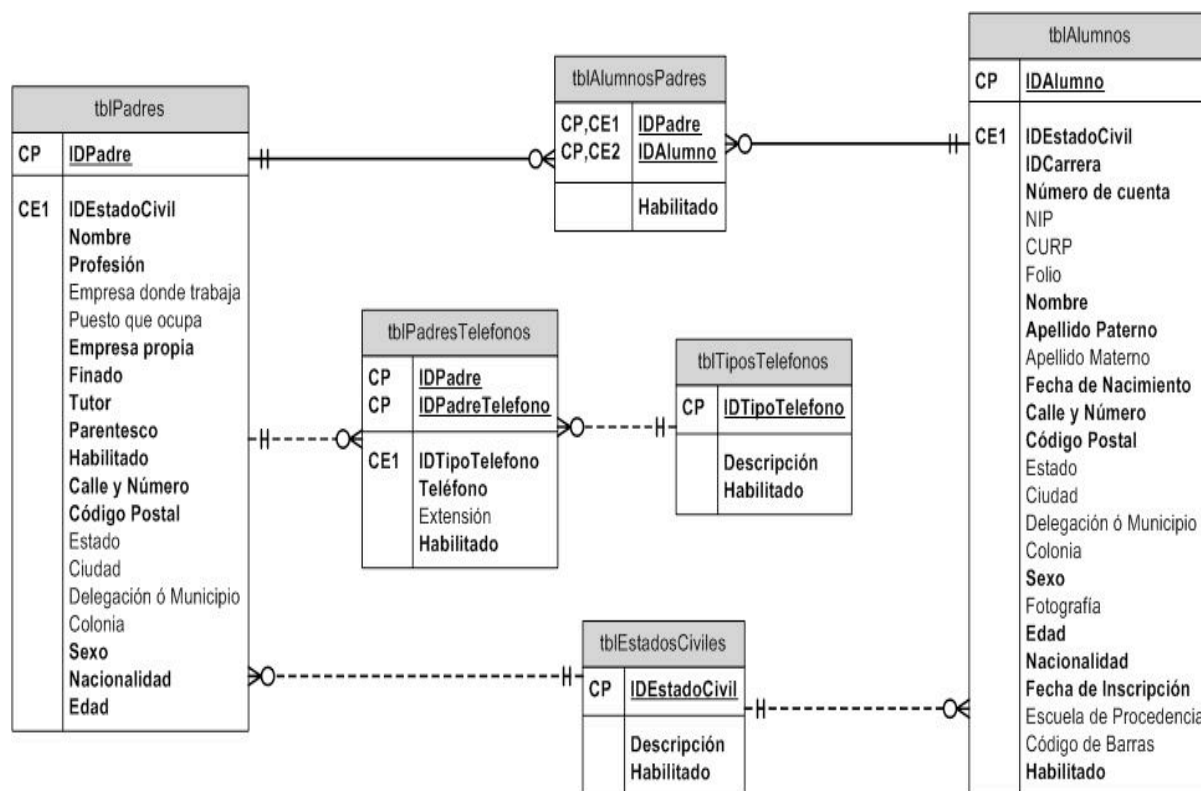


Figura 3.7. Modelo E-R del Módulo Alumno-Padre o Tutor.

## Alumnos Pagos

Para el módulo de alumnos-pagos, ver figura 3.8., presentamos el modelo E-R que muestra la relación entre el alumno y los pagos. Los pagos se refieren a los distintos tipos de pago, el monto, el monto del IVA, los recargos, descuentos, la fecha de realización, entre otros.

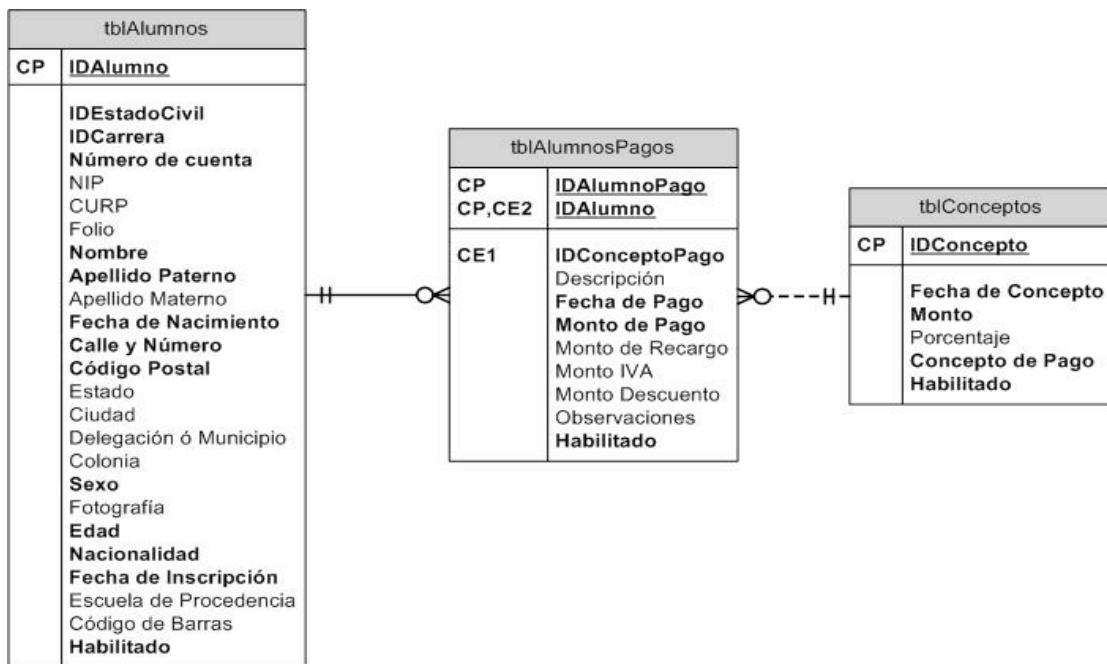


Figura 3.8. Módulo de Alumnos-Pagos.

## Profesores

El módulo de profesores, ver figura 3.9., está compuesto de varias entidades que integran la información suficiente y necesaria para el control, clasificación y el expediente del personal docente. El módulo de profesores comprende las siguientes entidades:

- Profesores. Contiene la información a datos personales, domicilio, cédula profesional, archivo de fotografía reciente, entre otros datos.
- Teléfonos. Números telefónicos donde el profesor pueda ser localizado.
- Especialidades. Descripción de las especialidades en el que el profesor esté capacitado.
- Grados Académicos. Se refiere al grado o grados obtenidos por el profesor.
- Mail. Se refiere a las diferentes direcciones electrónicas que tiene un profesor.
- Inasistencias. Maneja la estadística de inasistencias del profesor indicando la fecha y el motivo por la que ocurrió.
- Medicina General. Incluye un catálogo de enfermedades, tipos de sangre e información clínica que se relaciona con el profesor.

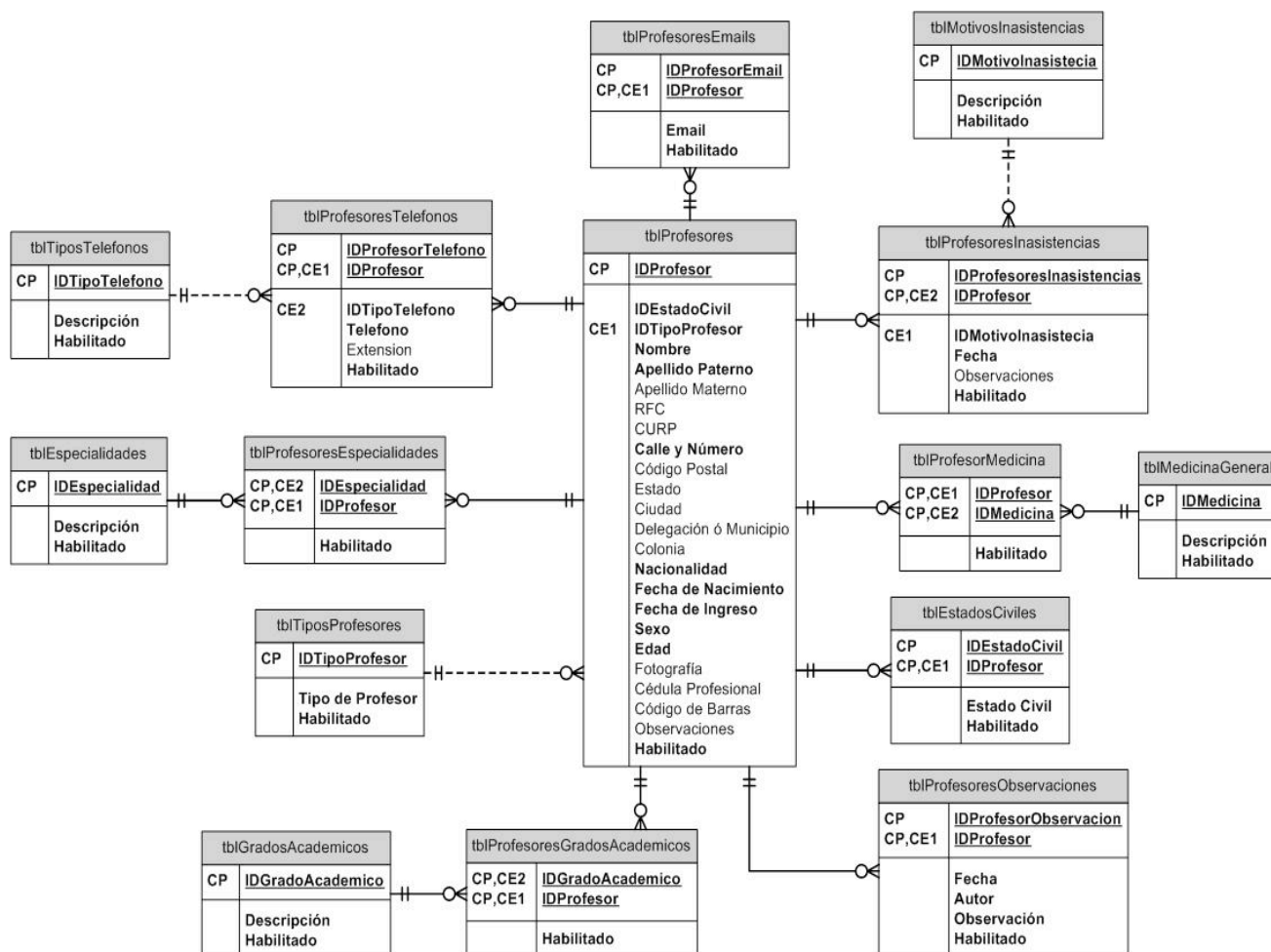


Figura 3.9. Modelo E-R del Módulo de Profesores.

## Planes de Estudio

El módulo de planes de estudio, ver figura 3.10, comprende las entidades siguientes:

- Plan de estudio. Contiene la descripción del plan, número de materias obligatorias, número de materias opcionales, fecha en que se aprobó, duración del plan de estudios, valor total de créditos obligatorios y opcionales.
- Materias. Contiene la descripción de la materia, tipo de la materia, si es de laboratorio o teórica, valor en créditos, cantidad máxima de alumnos que pueden ser inscritos, e indica la seriación entre materias si existe.

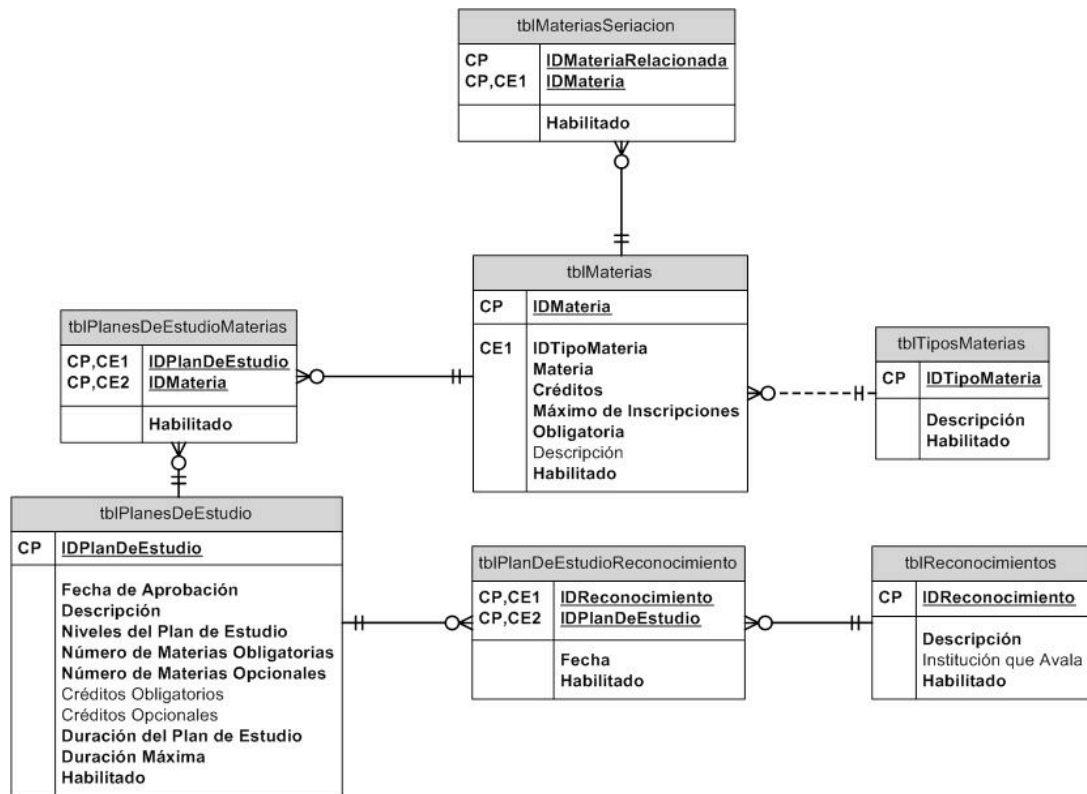


Figura 3.10. Modelo E-R del Módulo de Planes de Estudio.

Con base en la información presentada, estructuramos el diagrama lógico de la base de datos, que se muestra en la figura 3.11. En este diagrama se presentan las relaciones entre las entidades del sistema, alumnos, profesores, carreras, plan de estudios, materias, pagos, etcétera. En el diagrama se observa como el sistema integra la información referente a:

- Control de acceso y autenticación, en el que se define los permisos y privilegios para grupos y usuarios.
- Alumnos, en la cual se incluyen sus datos personales, de inscripción, reinscripción y pagos.
- Profesores, en el que se definen sus datos personales, control de incidencias, perfil y especialidad en la docencia.
- Plan de estudios se define la fecha en que se aprobó, la vigencia, la relación con las materias que le corresponde según lo acordado entre la universidad y la SEP.



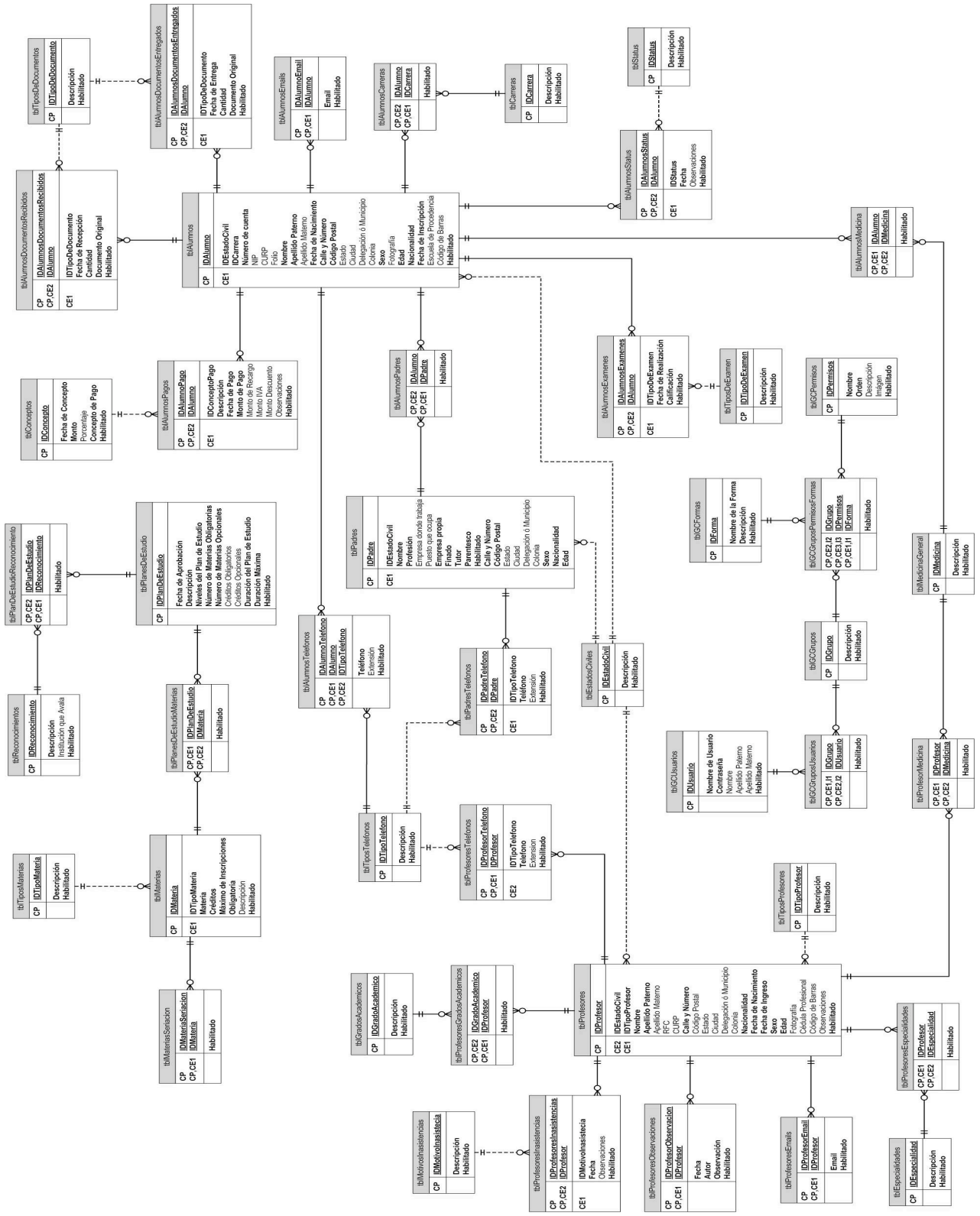


Figura 3.11. Modelo E-R del Sistema Administrador de Información.

En la base de datos se define los campos de información de los catálogos y tablas de trabajo, así como la integridad referencial, aplicando las primeras cuatro formas de normalización.

La base de datos comprende las áreas de servicios y administración escolar, el sistema tiene como finalidad administrar la información contenida en la base de datos de manera grafica, sencilla, funcional y en un ambiente agradable para el usuario. El diagrama entidad relación mostrado en la figura 3.11., nos permite observar todas las entidades que conforman el sistema de manera integral, para posteriormente describir cada parte, y observar que elementos contienen y como se interrelacionan.

En el modelo de la base de datos se manejó una nomenclatura convencional para hacerla entendible y mantener una coherencia en los términos de los datos.

La nomenclatura utilizada en el diagrama maneja el prefijo **tbl** en el nombre de cada tabla. Por ejemplo: tblalumnos, tblprofesores, etcétera.

El identificador de la llave primaria lleva el prefijo **id**, que proviene de las palabras **identificador de**. Por ejemplo idAlumno, idprofesor, etcétera. Las llaves primarias en su mayoría son representadas por valores numéricos enteros.

Con base en la información presentada, en el modelo de entidad-relación, procedemos a mostrar el informe general de las tablas que componen la base de datos del sistema de información, la información se describe en la tabla 3.1.

### Informe de Tablas

Tablas	Columnas	Claves Externas	Descripción
<b>tblAlumnos</b>	25	1	Tabla de Alumnos
<b>tblAlumnosCarreras</b>	3	2	Tabla relación alumnos-carreras
<b>tblAlumnos-DocumentosEntregados</b>	7	2	Tabla relación alumnos documentos
<b>tblAlumnos-DocumentosRecibidos</b>	7	2	Tabla relación alumnos-documentos
<b>tblAlumnosEmails</b>	4	1	Tabla que lista los correos electrónicos de los alumnos

Tabla 3.1. Informe de Tablas. (Continúa)



Tablas	Columnas	Claves Externas	Descripción
<b>tblAlumnosExámenes</b>	6	2	Tabla relación alumnos-Exámenes
<b>tblAlumnosMedicina</b>	3	2	Tabla relación alumnos-medicina
<b>tblAlumnosPadres</b>	3	2	Tabla relación alumnos-padres
<b>tblAlumnosPagos</b>	11	2	Tabla relación alumnos-pagos
<b>tblAlumnosStatus</b>	6	2	Tabla relación alumnos-status
<b>tblAlumnosTeléfonos</b>	6	2	Tabla relación alumnos-teléfonos
<b>tblCarreras</b>	3	0	Tabla de Carreras
<b>tblConceptos</b>	6	0	Catálogo de conceptos de pago
<b>tblEspecialidades</b>	3	0	Tabla de especialidades
<b>tblEstadosCiviles</b>	3	0	Catálogo de estado civil
<b>tblGCFormas</b>	4	0	Tabla que lista formas
<b>tblGCGrupos</b>	3	0	Tabla que lista los grupos de usuarios
<b>tblGCGrupos-PermisosFormas</b>	4	3	Tabla relación grupos-permisos de formas
<b>tblGCGruposUsuarios</b>	3	2	Tabla relación Grupos-Usuarios
<b>tblGCPermisos</b>	6	0	Tabla de permisos
<b>tblGCUsuarios</b>	7	0	Tabla de usuarios

*Tabla 3.1. Informe de Tablas. (Continúa)*

<b>Tablas</b>	<b>Columnas</b>	<b>Claves Externas</b>	<b>Descripción</b>
<b>tblGradosAcadémicos</b>	3	0	Tabla de Carreras
<b>tblMaterias</b>	8	1	Tabla catálogo de materias
<b>tblMedicinaGeneral</b>	3	0	Tabla catálogo de medicina general
<b>tblMotivosInasistencias</b>	3	0	Tabla de motivos de inasistencias
<b>tblPadres</b>	20	1	Tabla relación padres-alumnos
<b>tblPadresTeléfonos</b>	6	2	Tabla relación padres teléfonos
<b>tblPlanDeEstudio-Reconocimiento</b>	3	2	Tabla relación planes de estudio reconocimientos
<b>tblPlanesDeEstudio</b>	11	0	Table de planes de estudio
<b>tblPlanesDeEstudioMaterias</b>	3	2	Tabla relación Grados Académicos Materias
<b>tblProfesores</b>	24	2	Tabla de profesores
<b>tblProfesoresEmails</b>	4	1	Tabla de Correos electrónicos de Profesores
<b>tblProfesoresEspecialidades</b>	3	2	Tabla relación profesores especialidades
<b>tblProfesores-GradosAcadémicos</b>	3	2	Tabla puente profesores grados académicos

*Tabla 3.1. Informe de Tablas. (Continúa)*

Tablas	Columnas	Claves Externas	Descripción
<b>tblProfesores-Inasistencias</b>	6	2	Tabla profesores-inasistencias
<b>tblProfesores-Observaciones</b>	6	1	Tabla Profesores-Observaciones
<b>tblProfesoresTeléfonos</b>	6	2	Tabla que lista los teléfonos de los profesores
<b>tblProfesorMedicina</b>	3	2	Tabla relación Profesor-Medicina
<b>tblReconocimientos</b>	4	0	Tabla catálogo de reconocimientos
<b>tblStatus</b>	3	0	Tabla Catálogo Status
<b>tblTiposDeDocumentos</b>	3	0	Tabla tipos de documentos
<b>tblTiposDeExamen</b>	3	0	Tabla tipos de examen
<b>tblTiposMaterias</b>	3	0	Tabla catálogo que lista materias
<b>tblTiposProfesores</b>	3	0	Tabla de tipos de Profesores
<b>tblTiposTeléfonos</b>	3	0	Tabla de tipos de teléfonos
<b>tblMateriasSeriacion</b>	3	1	Tabla que lista las Materias Seriadadas

*Tabla 3.1. Informe de Tablas.*

### Detalle de Tablas

Con base en la información del informe de tablas, procedemos a detallar los campos que pertenecen a cada una de ellas y que conforma el sistema administrador de información.

*tblAlumnos*

Tabla de Alumnos, entidad principal que contiene información personal y de control del alumno, las columnas se describen en la tabla 3.2.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumno</b>	Integer	No	Identificador único del alumno
<b>IDEstadoCivil (CE)</b>	Integer	No	Clave externa para acceder al catálogo de estado civil
<b>IDCarrera</b>	Integer	No	Identificador de carrera
<b>Número de cuenta</b>	Varchar(10)	No	Número que identifica al alumno para realizar los trámites, es una composición de información de acuerdo a su año de ingreso, tipo de carrera y un consecutivo de cuatro dígitos
<b>NIP</b>	Varchar(30)	Sí	Número de identificación personal para permitir el acceso de información al alumno
<b>CURP</b>	Varchar(25)	Sí	Clave Única de Registro de Población
<b>Folio</b>	Varchar(30)	Sí	Clave que indica la localización del expediente del alumno
<b>Nombre</b>	Varchar(50)	No	Nombre del alumno
<b>Apellido Paterno</b>	Varchar(50)	No	Primer apellido del alumno
<b>Apellido Materno</b>	Varchar(50)	Sí	Segundo apellido del alumno si lo tiene
<b>Fecha de Nacimiento</b>	DateTime	No	Fecha en que el alumno nació según su acta de nacimiento

*Tabla 3.2. Columnas de la entidad tblAlumnos. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>Calle y Número</b>	Varchar(100)	No	Nombre de la calle y número de donde vive el alumno
<b>Código Postal</b>	Varchar(10)	No	Número de código postal de la zona
<b>Estado</b>	Varchar(255)	Sí	Estado de la República Mexicana
<b>Ciudad</b>	Varchar(255)	Sí	Ciudad donde vive el alumno actualmente
<b>Delegación ó Municipio</b>	Varchar(255)	Sí	Delegación donde vive el alumno actualmente
<b>Colonia</b>	Varchar(255)	Sí	Nombre de la colonia donde vive el alumno
<b>Sexo</b>	CHAR(1)	No	Valor de H para hombre y M para mujer
<b>Fotografía</b>	Varchar(255)	Sí	Ruta de la Imagen fotográfica del alumno
<b>Edad</b>	SMALLINT	No	Número que describe su edad
<b>Nacionalidad</b>	CHAR(1)	No	N para nacional y E para extranjero
<b>Fecha de Inscripción</b>	DateTime	No	Fecha en que se incorporó como alumno a la institución
<b>Escuela de Procedencia</b>	Varchar(255)	Sí	Nombre de la escuela anterior de procedencia
<b>Código de Barras</b>	Varchar(50)	Sí	Cadena que representa el código de barras

*Tabla 3.2. Columnas de la entidad tblAlumnos. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.2. Columnas de la entidad *tblAlumnos*.

*tblAlumnosCarreras*

Ésta es una tabla puente que ayuda a resolver la relación de que un alumno puede tener una o muchas carreras o de que una carrera debe tener uno o muchos alumnos, se describe en la figura 3.3.

Columnas	Tipo de Dato	Nulo	Descripción
IDAlumno (CE)	Integer	No	Identificador único del alumno
IDCarrera (CE)	Integer	No	Identificador único de carrera
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.3. Columnas de la entidad *tblAlumnosCarreras*.

*tblAlumnosDocumentosEntregados*

Esta tabla se define para controlar la entrega de los documentos que la escuela da a los alumnos como respuesta de sus trámites, sus columnas se describen en la tabla 3.4.

Columnas	Tipo de Dato	Nulo	Descripción
IDAlumnosDocumentosEntregados	Integer	No	Identificador único alumnos-documentos-entregados

Tabla 3.4. Columnas de la entidad *tblAlumnosDocumentosEntregados*. (Continúa)

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumno (CE)</b>	Integer	No	Clave externa del identificador único del alumno
<b>IDTipoDeDocumento (CE)</b>	Integer	No	Clave externa del identificador de documento
<b>Fecha de Entrega</b>	DateTime	No	Fecha de la entrega del documento
<b>Cantidad</b>	Integer	No	Número de documentos entregados por parte de la escuela
<b>Documento Original</b>	Bit	No	Identifica si es el documento es original o no
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.4. Columnas de la entidad `tblAlumnosDocumentosEntregados`.*

#### *tblAlumnosDocumentosRecibidos*

Esta tabla se define para controlar la entrega de los documentos que la escuela requiere de los alumnos, para todos los trámites del mismo. Sus columnas se describen en la tabla 3.5.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumnos Documentos Recibidos</b>	Integer	No	Identificador único de alumnos-documentos-recibidos
<b>IDAlumno</b>	Integer	No	Clave externa del identificador del alumno

*Tabla 3.5. Columnas de la entidad `tblAlumnosDocumentosRecibidos`. (Continúa).*

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoDe Documento</b>	Integer	No	Clave externa del tipo de documento
<b>Fecha de Recepción</b>	DateTime	No	Fecha de recepción del documento
<b>Cantidad</b>	Integer	No	Número de documentos recibidos por parte de la escuela
<b>Documento Original</b>	Bit	No	Identifica si es el documento es original o no
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.5. Columnas de la entidad tblAlumnosDocumentosRecibidos.*

*tblAlumnosEmails*

Esta tabla se definió para poder tener el registro de los correos electrónicos que pudiera tener el alumno, las columnas se describen en la tabla 3.6.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumnoEmail</b>	Integer	No	Identificador único de alumno email
<b>IDAlumno (CE)</b>	Integer	No	Clave externa del identificador de alumno
<b>Email</b>	Varchar(255)	No	Nombre del correo electrónico
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.6. Columnas de la entidad tblAlumnosEmails..*



*tblAlumnosExámenes*

Esta tabla se definió para poder llevar el registro de los exámenes de admisión; sin embargo, tiene la capacidad para registrar todos los exámenes del alumno, las columnas se describen en la tabla 3.7.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumnosExámenes</b>	Integer	No	Identificador único de alumnos exámenes
<b>IDAlumno (CE)</b>	Integer	No	Clave externa de alumno
<b>IDTipoDeExamen (CE)</b>	Integer	No	Clave externa del tipo de examen
<b>Fecha de Realización</b>	DateTime	No	Fecha de realización del examen
<b>Calificación</b>	Decimal(2,2)	No	Calificación del examen
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.7. Columnas de la entidad tblAlumnosExámenes.*

*tblAlumnosMedicina*

Esta es una tabla puente que ayuda a resolver la relación de que un alumno puede tener una o muchas referencias médicas o de que una referencia médica puede tener uno o muchos alumnos, las columnas se describen en la tabla siguiente:

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumno (CE)</b>	Integer	No	Clave externa de alumno
<b>IDMedicina (CE)</b>	Integer	No	Clave externa de medicina

*Tabla 3.8. Columnas de la entidad tblAlumnosMedicina. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.8. Columnas de la entidad *tblAlumnosMedicina*.

*tblAlumnosPadres*

Esta es una tabla puente que ayuda a resolver la relación de que un alumno debe tener uno o muchos profesores, o de que un profesor debe tener uno o muchos alumnos, las columnas se describen en la tabla 3.9.

Columnas	Tipo de Dato	Nulo	Descripción
IDAlumno (CE)	Integer	No	Clave externa de alumno
IDPadre (CE)	Integer	No	Clave externa de padre
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.9. Columnas de la entidad *tblAlumnosPadres*.

*tblAlumnosPagos*

Esta tabla se define para controlar el pago de los alumnos, las columnas se describen en la tabla siguiente:

Columnas	Tipo de Dato	Nulo	Descripción
IDAlumnoPago	Integer	No	Clave única de identificación de pago
IDAlumno (CE)	Integer	No	Clave externa de alumno

Tabla 3.10. Columnas de la entidad *tblAlumnosPagos*. (Continúa)

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDConceptoPago (CE)</b>	Integer	No	Clave externa de concepto pago
<b>Descripción</b>	Varchar(255)	No	Descripción del pago efectuado
<b>Fecha de Pago</b>	DateTime	No	Fecha en que se recibe el pago
<b>Monto de Pago</b>	Currency	No	Cantidad en pesos del pago
<b>Monto de Recargo</b>	Currency	Sí	Cantidad del recargo por incurrir en alguna falta de acuerdo a las políticas de la escuela
<b>Monto IVA</b>	Currency	Sí	Cantidad de IVA
<b>Monto Descuento</b>	Currency	Sí	Monto de descuento por alguna Beca o aclaración
<b>Observaciones</b>	Text(10)	Sí	Observaciones especiales referentes al pago
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.10. Columnas de la entidad tblAlumnosPagos.*

*tblAlumnosStatus.*

Esta tabla se define para integrar las observaciones especiales que pudiera tener el alumno, las columnas se describen en la tabla 3.11.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumnosStatus</b>	Integer	No	Clave única de observación

*Tabla 3.11. Columnas de la entidad tblAlumnosStatus. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumno (CE)</b>	Integer	No	Clave externa de alumno
<b>IDStatus (CE)</b>	Integer	No	Clave externa de status
<b>Fecha</b>	DateTime	No	Fecha de alta de la observación
<b>Observaciones</b>	Text(10)	Sí	Observación especial
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.11. Columnas de la entidad tblAlumnosStatus.*

*tblAlumnosTeléfonos*

Esta tabla se define para registrar todos los teléfonos que pudieran tener los alumnos, las columnas se describen en la tabla 3.12.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDAlumnoTeléfono</b>	Integer	No	Clave única de alumno teléfono
<b>IDAlumno (CE)</b>	Integer	No	Clave externa de alumno
<b>Teléfono</b>	Varchar(50)	No	Teléfono del alumno
<b>Extensión</b>	SMALLINT	Sí	Número de extensión

*Tabla 3.12. Columnas de la entidad tblAlumnosTeléfonos. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro
IDTipoTeléfono (CE)	Integer	No	Clave externa del tipo de teléfono

*Tabla 3.12. Columnas de la entidad tblAlumnosTeléfonos.*

#### *tblCarreras*

Tabla que define la relación entre el alumno y la o las carreras que pueda tener dentro de la institución, las columnas se describen en la tabla 3.13.

Columnas	Tipo de Dato	Nulo	Descripción
IDCarrera	Integer	No	Clave única de carrera
Descripción	Varchar(150)	No	Descripción de la carrera
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.13. Columnas de la entidad tblCarreras.*

#### *tblConceptos*

Tabla que define la relación entre el alumno y la o las carreras que pueda tener dentro de la institución, los columnas se describen en la tabla 3.14.

Columnas	Tipo de Dato	Nulo	Descripción
IDConcepto	Integer	No	Clave única del concepto

*Tabla 3.14. Columnas de la entidad Conceptos. (Continúa).*

Columnas	Tipo de Dato	Nulo	Descripción
<b>Fecha de Concepto</b>	DateTime	No	Fecha de alta del concepto
<b>Monto</b>	Currency	No	Monto mínimo por el pago de este concepto
<b>Porcentaje</b>	Decimal(3,2)	Sí	Valor flotante que representa el porcentaje del concepto de pago
<b>Concepto de Pago</b>	Varchar(10)	No	Descripción del pago
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.14. Columnas de la entidad tblConceptos.*

*tblEspecialidades*

Tabla que define las especialidades que el profesor pudiera tener, las columnas se describen en la tabla 3.15.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDEspecialidad</b>	Integer	No	Clave única de la identificación de la especialidad
<b>Descripción</b>	Varchar(150)	No	Descripción de la especialidad del profesor
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.15. Columnas de la entidad tblEspecialidades.*

*tblEstadosCiviles*

Catálogo que especifica los diferentes estados civiles que a las personas se les puede asignar, se describen los columnas en la tabla 3.16.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDEstadoCivil</b>	Integer	No	Clave única de estado civil
<b>Descripción</b>	Varchar(50)	No	Descripción del estado civil
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.16. Columnas de la entidad tblEstadosCiviles.*

*tblGCFormas*

Esta tabla define a las formas o pantallas que existen en el sistema de información y que estarán disponibles para los grupos de usuarios que se definan, se describen las columnas en la tabla 3.17.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDForma</b>	Integer	No	Clave única de forma
<b>Nombre de la Forma</b>	Varchar(255)	No	Nombre de la forma
<b>Descripción</b>	Varchar(255)	No	Descripción de la forma
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.17. Columnas de la entidad GCFormas.*

*tblGCGrupos.*

Esta tabla se define para especificar todos los grupos a los que se les permitirá el acceso al sistema de información, se describen las columnas en la tabla 3.18.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDGrupo</b>	Integer	No	Clave única de grupo
<b>Descripción</b>	Varchar(255)	No	Descripción del grupo
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.18. Columnas de la entidad tblGCGrupos.*

*tblGCGruposPermisosFormas*

Esta es una tabla puente que ayuda a resolver la relación de que un grupo debe tener una o muchas formas, o de que una forma debe definirse para uno o muchos grupos. Así también, un grupo debe tener uno o muchos permisos o de que un permiso debe ser asignado para uno o muchos grupos, se describen las columnas en la tabla 3.19.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDGrupo (CE,I2)</b>	Integer	No	Clave externa de grupo
<b>IDPermisos (CE,I3)</b>	Integer	No	Clave externa de permisos
<b>IDForma (CE,I1)</b>	Integer	No	Clave externa de forma
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.19. Columnas de la entidad tblGCGruposPermisosFormas.*



*tblGCGruposUsuarios*

Ésta es una tabla puente que ayuda a resolver la relación de que un grupo debe tener uno o muchos usuarios, o de que un usuario debe asignarse para uno o muchos grupos, se describen las columnas o los nombres de los campos en la tabla 3.20.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDGrupo (CE,I1)</b>	Integer	No	Clave externa de grupo
<b>IDUsuario (CE,I2)</b>	Integer	No	Clave externa de usuario
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.20. Columnas de la entidad tblGCGruposUsuarios.*

*tblGCPermisos*

Esta tabla se define para especificar si el usuario tiene permiso para acceder a alguna de las formas definidas en el sistema de información, se describen las columnas en la tabla 3.21.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPermisos</b>	Integer	No	Clave única de permiso
<b>Nombre</b>	Varchar(255)	No	Nombre del permiso
<b>Orden</b>	Integer	No	Especificación del permiso
<b>Descripción</b>	Varchar(255)	Si	Descripción del permiso
<b>Imagen</b>	Varchar(255)	Si	Ruta física de la imagen que representa el nombre de acción del registro
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.21. Columnas de la entidad tblGCPermisos.*

*tblGCUusuarios*

Esta tabla se define para integrar a todos los usuarios que tendrán acceso al sistema de información, se describen las columnas en la tabla 3.22.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDUsuario</b>	Integer	No	Clave única de usuario
<b>Nombre de Usuario</b>	Varchar(10)	No	Nombre corto del usuario para acceso
<b>Contraseña</b>	Varchar(10)	No	Contraseña del usuario para acceso
<b>Nombre</b>	Varchar(255)	Si	Nombre de usuario
<b>Apellido Paterno</b>	Varchar(255)	Si	Apellido paterno
<b>Apellido Materno</b>	Varchar(255)	Si	Apellido materno
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.22. Columnas de la entidad tblGCUusuarios.*

*tblGradosAcadémicos*

Esta tabla se define para integrar los grados académicos existentes que pueden ser asignados a los profesores, la descripción de las columnas se muestran en la tabla 3.23.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDGradoAcademico</b>	CHAR(10)	No	Clave única de grado academico

*Tabla 3.23. Columnas de la entidad tblGCUusuarios. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
Descripción	Varchar(150)	No	Descripción del grado academico
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.23. Columnas de la entidad *tblGUsuarios*.

### *tblMaterias*

Esta tabla se define para recabar todas las materias existentes en todos los planes de estudio, las columnas se describen en la tabla 3.24.

Columnas	Tipo de Dato	Nulo	Descripción
IDMateria	Integer	No	Clave única de materia
IDTipoMateria (CE)	Integer	No	Clave externa de tipo de materia
Materia	Varchar(255)	No	Nombre de la materia
Créditos	SMALLINT	No	Especificación de créditos
Máximo de Inscripciones	SMALLINT	No	Número máximo de inscripciones a la materia
Obligatoria	Bit	No	Especifica si la materia es obligatoria u optativa
Descripción	Varchar(255)	Sí	Descripción de la materia
Habilitado	Bit	No	Este campo sirve para establecer una baja lógica del registro

Tabla 3.24. Columnas de la entidad *tblMaterias*.

*tblMedicinaGeneral*

Catálogo medicina general, lista de padecimientos, tipos de sangre, etcétera, las columnas de la entidad se describen en la siguiente tabla.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDMedicina</b>	Integer	No	Identificador único, valor consecutivo
<b>Descripción</b>	Varchar(255)	No	Descripción
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.25. Columnas de la entidad tblMedicinaGeneral.*

*tblMotivosInasistencias*

Catálogo general que describe los posibles motivos de inasistencia, tanto para profesores como alumnos, las columnas se describen en la tabla 3.26.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDMotivolnasistec ia</b>	Integer	No	Identificador único, valor consecutivo
<b>Descripción</b>	Varchar(255)	No	Descripción de los posibles motivos de inasistencia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.26. Columnas de la entidad tblMotivosInasistencias.*

*tblPadres*

Catálogo de padres, contiene la información del padre o tutor de cada alumno, datos personales, datos laborales y domicilio, las columnas se describen en la tabla 3.27.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPadre</b>	Integer	No	Identificador único del padre o tutor
<b>IDEstadoCivil (CE)</b>	Integer	No	Llave foránea que proviene del catálogo estado civil
<b>Nombre</b>	Varchar(255)	No	Nombre completo del padre o tutor
<b>Profesión</b>	Varchar(255)	No	Actividad a la que se dedica.
<b>Empresa donde trabaja</b>	Varchar(255)	Sí	Nombre de la empresa donde presta sus servicios
<b>Puesto que ocupa</b>	Varchar(255)	Sí	Cargo que ocupa en la empresa donde labora
<b>Empresa propia</b>	CHAR(1)	No	Si cuenta con una empresa o negocio es verdadero, en caso contrario es falso
<b>Finado</b>	CHAR(1)	No	Indica si el padre ha muerto o no
<b>Tutor</b>	Bit	No	En caso de que se trate del tutor será verdadero, en otro caso contrario es falso
<b>Parentesco</b>	CHAR(1)	No	Si se trata de un pariente directo es verdadero, en otro caso, es falso
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro
<b>Calle y Número</b>	Varchar(100)	No	Nombre de la calle y número exterior e interior de donde vive el alumno

*Tabla 3.27. Columnas de la entidad tblPadres. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>Código Postal</b>	Varchar(10)	No	Número de código postal de la zona
<b>Estado</b>	Varchar(255)	Sí	Guarda la delegación donde vive el alumno actualmente
<b>Ciudad</b>	Varchar(255)	Sí	Ciudad donde vive el alumno actualmente
<b>Delegación ó Municipio</b>	Varchar(255)	Sí	Fecha en que se incorporó como alumno a la institución
<b>Colonia</b>	Varchar(255)	Sí	Nombre de la colonia donde se localiza su domicilio
<b>Sexo</b>	CHAR(1)	No	Si es hombre 1 si es mujer 0
<b>Nacionalidad</b>	CHAR(1)	No	Si se trata de un extranjero es verdadero de lo contrario es falso, se trata de un alumno de nacionalidad mexicana
<b>Edad</b>	SMALLINT	No	Número que describe su edad

*Tabla 3.27. Columnas de la entidad tblPadres.*

*tblPadresTeléfonos*

Tabla en donde se almacena los números telefonicos correspondientes a cada profesor, las columnas se describen en la tabla 3.28.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPadreTeléfono</b>	Integer	No	Identificador Consecutivo
<b>IDPadre (CE)</b>	Integer	No	Llave foránea de la tabla de padres

*Tabla 3.28. Columnas de la entidad tblPadresTeléfonos. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoTeléfono (CE)</b>	Integer	No	Llave foránea del catálogo de tipo de teléfonos
<b>Teléfono</b>	Varchar(50)	No	Número telefónico
<b>Extensión</b>	SMALLINT	Sí	Extensión de ser necesario
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.28. Columnas de la entidad **tblPadresTeléfonos**.*

*tblPlanDeEstudioReconocimiento*

Tabla que muestra la relación entre el plan de estudio y el catálogo de reconocimientos, las columnas se describen en la tabla siguiente.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPlanDeEstudio (CE)</b>	Integer	No	Llave foránea, identificador del plan de estudios
<b>IDReconocimiento (CE)</b>	Integer	No	Llave foránea, identificador de la descripción del reconocimiento
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.29. Columnas de la entidad **tblPlanDeEstudioReconocimiento**.*

*tbIPlanesDeEstudio*

Catálogo que describe el plan de estudios, así como información de gran relevancia, las columnas se describen en la tabla 3.30.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPlanDeEstudio</b>	Integer	No	Identificador único al plan de estudios
<b>Fecha de Aprobación</b>	DateTime	No	Fecha en que fue aprobado el plan de estudios por la SEP
<b>Descripción</b>	Varchar(255)	No	Descripción acerca del plan de estudios
<b>Niveles del Plan de Estudio</b>	SMALLINT	No	Cantidad de niveles de los que está compuesto del plan de estudios
<b>Número de Materias Obligatorias</b>	SMALLINT	No	Valor del número de materias obligatorias
<b>Número de Materias Opcionales</b>	SMALLINT	No	Valor del número de materias opcionales
<b>Créditos Obligatorios</b>	SMALLINT	Si	Cantidad total de créditos obligatorios
<b>Créditos Opcionales</b>	SMALLINT	Si	Cantidad de créditos Opcionales
<b>Duración del Plan de Estudio</b>	SMALLINT	No	Vigencia del plan de estudios
<b>Duración Máxima</b>	SMALLINT	No	Cantidad de años que será válido el plan de estudios
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.30. Columnas de la entidad tbIPlanesDeEstudio.*



*tblPlanesDeEstudioMaterias*

Tabla puente que relaciona el catálogo de planes de estudio con el catálogo de materias, a través de las llaves principales, las columnas se describen en la tabla 3.31.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDPlanDeEstudio</b>	Integer	No	Llave foránea del catálogo de la tabla plan de estudio
<b>IDMateria</b>	Integer	No	Llave foránea del catálogo de la tabla plan de estudio
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.31. Columnas de la entidad tblPlanesDeEstudioMaterias.*

*tblProfesores*

Contiene la información básica pero primordial para el registro de profesores. Se almacena sus datos personales, domicilio, entre otros, las columnas se describen en la tabla 3.32.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesor</b>	Integer	No	Llave principal que identifica a los profesores. Es un número consecutivo
<b>IDEstadoCivil (CE)</b>	Integer	No	Llave foránea, que indica el estado civil del profesor
<b>IDTipoProfesor (CE)</b>	Integer	No	Llave foránea que indica el tipo de profesor de acuerdo al catálogo de tipos de profesor
<b>Nombre</b>	Varchar(50)	No	Nombre(s) del profesor
<b>Apellido Paterno</b>	Varchar(50)	No	Primer Apellido del profesor

*Tabla 3.32. Columnas de la entidad tblProfesores. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>Apellido Materno</b>	Varchar(50)	Si	Segundo Apellido s i lo tiene
<b>RFC</b>	Varchar(20)	Si	Registro Federal de Causantes
<b>CURP</b>	Varchar(20)	Si	Clave Única de Registro de Población
<b>Calle y Número</b>	Varchar(100)	No	Nombre de la calle y número exterior e interior de donde vive el profesor
<b>Código Postal</b>	Varchar(10)	Si	Número del código postal donde reside
<b>Estado</b>	Varchar(255)	Si	Estado donde vive actualmente el profesor
<b>Ciudad</b>	Varchar(255)	Si	Ciudad donde vive el alumno actualmente
<b>Delegación ó Municipio</b>	Varchar(255)	Si	Guarda la delegación donde vive el alumno actualmente
<b>Colonia</b>	Varchar(255)	Si	Nombre de la colonia donde vive el profesor
<b>Nacionalidad</b>	CHAR(1)	No	Si se trata de un extranjero es verdadero de lo contrario es falso, se trata de un alumno de nacionalidad mexicana
<b>Fecha de Nacimiento</b>	DateTime	No	Fecha de Nacimiento
<b>Fecha de Ingreso</b>	DateTime	No	Fecha en que ingresó a la institución como profesor

Tabla 3.32. Columnas de la entidad tblProfesores. (Continúa)

Columnas	Tipo de Dato	Nulo	Descripción
<b>Sexo</b>	CHAR(1)	No	Si es hombre 1 si es mujer 0
<b>Edad</b>	SMALLINT	No	Edad del profesor
<b>Fotografía</b>	Varchar(255)	Si	Ruta del archivo imagen donde se encuentra la fotografía del profesor
<b>Cédula Profesional</b>	Varchar(50)	Si	Identificador de su cédula profesional
<b>Código de Barras</b>	Varchar(50)	Si	Código de Barras de su credencial
<b>Observaciones</b>	Text(10)	Si	Breve anotación sobre algo relevante del profesor
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.32. Columnas de la entidad tblProfesores.*

#### *tblProfesoresEmails*

Tabla que almacena las dirección o direcciones de correo electrónico de los profesores, las columnas se describen en la tabla 3.33.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesorEmail</b>	Integer	No	Identificador único, número consecutivo
<b>IDProfesor</b>	Integer	No	Llave foránea de la tabla profesor
<b>Email</b>	Varchar(255)	No	Dirección de correo electrónico
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.33. Columnas de la entidad tblProfesoresEmails.*

*tblProfesoresEspecialidades*

Relación entre el profesor y sus especialidades según el catálogo de de especialidades, las columnas se describen en la tabla 3.34.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesor</b>	Integer	No	Llave foránea de la tabla Profesores
<b>IDEspecialidad</b>	Integer	No	Llave foránea de la tabla especialidades
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.34. Columnas de la entidad tblProfesoresEspecialidades.*

*tblProfesoresGradosAcadémicos*

Tabla que relaciona al profesor con uno o varios grados académicos, las columnas se describen en la tabla 3.35.

Columnas	Tipo de Dato	Nulo	Descripción
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro
<b>IDGradoAcademico</b>	CHAR(10)	No	Llave foránea de la tabla grado académico
<b>IDProfesor</b>	Integer	No	Llave foránea de la tabla profesor

*Tabla 3.35. Columnas de la entidad tblProfesoresGradosAcadémicos.*

*tblProfesoresInasistencias*

Contiene la información acerca de la relación entre profesores y el catálogo de motivos de inasistencia, llevando un registro de inasistencias, las columnas se describen en la tabla 3.36.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesoresInasistencias</b>	Integer	No	Consecutivo
<b>IDProfesor</b>	Integer	No	Llave foránea de la tabla de profesores
<b>IDMotivoInasistencia</b>	Integer	No	Llave foránea del catálogo de inasistencias
<b>Fecha</b>	DateTime	No	Fecha en que se efectuó la inasistencia
<b>Observaciones</b>	Text(10)	Sí	Breve descripción en caso de existir una aclaración
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.36. Columnas de la entidad tblProfesoresInasistencias.*

*tblProfesoresObservaciones*

Tabla donde se registran las observaciones hechas a la planta de profesores de manera personal, las columnas se describen en la tabla 3.37.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesor-Observacion</b>	Integer	No	Consecutivo
<b>IDProfesor (CE)</b>	Integer	No	Llave foránea de la tabla de profesores
<b>Fecha</b>	DateTime	No	Fecha en que se realice la observación
<b>Autor</b>	Varchar(150)	No	Nombre de quien realice la observación

*Tabla 3.37. Columnas de la entidad tblProfesoresObservaciones. (Continúa)*

Columnas	Tipo de Dato	Nulo	Descripción
<b>Observación</b>	Text(10)	No	Varchar que describe la observación
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.37. Columnas de la entidad tblProfesoresObservaciones.*

*tblProfesoresTeléfonos*

Tabla relación entre el profesor y el teléfono o teléfonos que le pertenecen, las columnas se describen en la tabla 3.38.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesorTeléfono</b>	Integer	No	Consecutivo
<b>IDProfesor (CE)</b>	Integer	No	Llave foránea de la tabla de profesores
<b>IDTipoTeléfono (CE)</b>	Integer	No	Llave foránea del catálogo de tipo de teléfonos
<b>Teléfono</b>	Varchar(50)	No	Número telefónico
<b>Extension</b>	SMALLINT	Si	Extensión en caso de ser necesaria
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.38. Columnas de la entidad tblProfesoresTeléfonos.*

*tblProfesorMedicina*

Tabla que contiene la relación entre el profesor y el catálogo medicina general, las columnas se describen en la tabla 3.39.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDProfesor (CE)</b>	Integer	No	Llave foránea de la tabla de profesores
<b>IDMedicina (CE)</b>	Integer	No	Llave foránea del catálogo medicina general
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.39. Columnas de la entidad **tblProfesorMedicina**.*

### *tblReconocimientos*

Describe los reconocimientos que tiene el profesor en el ámbito de conocimientos y enseñanza, las columnas se describen en la tabla 3.40.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDReconocimiento</b>	Integer	No	Identificador único de los tipos de teléfono.
<b>Descripción</b>	Varchar(255)	No	Descripción de los tipos de materia.
<b>Institución que Avala</b>	Varchar(255)	Si	Contiene el nombre de la institución que otorgó el reconocimiento.
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.40. Columnas de la entidad **tblReconocimientos**.*

### *tblStatus*

Catálogo general, que contiene el listado de status de los alumnos, los campos o nombres de columnas se describen en la tabla 3.41.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDStatus</b>	Integer	No	Identificador único del catálogo
<b>Descripción</b>	Varchar(255)	No	Descripción de los tipos de materia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.41. Columnas de la entidad tblStatus.*

*tblTiposDeDocumentos.*

Catálogo general, que lista los tipos de documento que recibe o entrega la universidad, las columnas se describen en la tabla 3.42.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoDeDocumento</b>	Integer	No	Identificador único de los tipos de teléfono
<b>Descripción</b>	Varchar(255)	No	Descripción de los tipos de materia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.42. Columnas de la entidad tblTiposDeDocumentos.*

*tblTiposDeExamen*

Catálogo general, que lista los tipos de examen que alumno realiza en el transcurso de su vida curricular dentro de la universidad, las columnas se describen en la tabla 3.43.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoDeExamen</b>	Integer	No	Identificador único de los tipos de teléfono
<b>Descripción</b>	Varchar(255)	No	Descripción de los tipos de materia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.43. Columnas de la entidad tblTiposDeExamen.*



*tblTiposMaterias*

Catálogo general, que lista los tipos de materias, las columnas se describen en la tabla 3.44.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoMateria</b>	Integer	No	Identificador único de los tipos de teléfono
<b>Descripción</b>	Varchar(255)	No	Descripción de los tipos de materia
<b>Habilitado</b>	Bit	No	Estatus del registro

*Tabla 3.44. Columnas de la entidad tblTiposMaterias.*

*tblTiposProfesores*

Catálogo general, que lista los tipos de profesor que se encuentran dentro de la universidad, las columnas se describen en la tabla 3.45.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoProfesor</b>	Integer	No	Identificador único de los tipos de teléfono
<b>Descripción</b>	Varchar(150)	No	Descripción de los tipos de profesor.
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.46. Columnas de la entidad tblTiposProfesores.*

*tblTiposTeléfonos*

Catálogo general de los tipos de medios de comunicación via telefonica con los que se puede localizar a un profesor o alumno, las columnas se describen en la tabla 3.47.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDTipoTeléfono</b>	Integer	No	Identificador único de los tipos de teléfono
<b>Descripción</b>	Varchar(50)	No	Descripción de los temas de acuerdo a la materia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.47. Columnas de la entidad tblTiposTeléfonos.*

*tblMateriasSeriacion*

Tabla que relaciona las materias que tienen seriación hacia abajo, las columnas se describen en la tabla 3.48.

Columnas	Tipo de Dato	Nulo	Descripción
<b>IDMateriaSeriación</b>	Integer	No	Identificador único
<b>IDMateria (CE)</b>	Integer	No	Identificador de materia
<b>Habilitado</b>	Bit	No	Este campo sirve para establecer una baja lógica del registro

*Tabla 3.48. Columnas de la entidad tblMateriasSeriacion.*

Considerando el análisis y el diseño del sistema, en el capítulo siguiente se muestra la estructura del generador de código. Se explicará las distintas etapas que comprende la construcción de la herramienta que se usará para el desarrollo del Sistema Administrador de Información.

# 4

## Generador de Código

---

Las herramientas de programación permiten a los desarrolladores de software editar, interpretar, compilar, probar y depurar, todos los programas fuentes de una aplicación o un sistema, además de ayudar a la detección y comparación de errores, todo con la finalidad de que cada vez sea más fácil, rápido y agradable el desarrollo de los programas. En este capítulo se presenta la definición de la funcionalidad, el modelado de datos y objetos, las estructuras de datos y el desarrollo e implementación de la herramienta de generación de código que se usará para el desarrollo del Sistema Administrador de Información.

Existen diversos tipos de herramientas, desde editores y compiladores. En la actualidad están disponibles herramientas de programación orientadas a los objetos y herramientas y lenguajes de programación de cuarta generación, ambos son de las tecnologías más actuales en la ingeniería de software y se componen de una gran variedad de productos, los cuales hacen que cada vez menos código sea escrito manualmente.

Además, el objetivo final de la Ingeniería de Software Asistida por Computadora (*CASE*) es la generación del código automático, esto es, la representación de sistemas a un nivel de abstracción más alto que el de los lenguajes de programación convencionales. Idealmente, estas herramientas de generación de código no sólo traducirán la descripción de un sistema a un programa operativo, sino que también

ayudarán a verificar la corrección de la especificación del sistema, de tal forma que la salida resultante satisfaga los requisitos del usuario.

En términos generales un generador de código es un Software que genera programas de aplicación a partir de descripciones del problema, en lugar de una programación detallada, por lo cual requiere que el usuario defina de forma completa las acciones y funciones deseadas por su aplicación.

Los generadores de programas son entornos de programación que generan aplicaciones completas y no necesariamente generan el código correspondiente. Estos entornos pueden, a partir de una base de datos, generar vistas, pantallas de consulta y actualización, reportes, etc.; además de ofrecer múltiples alternativas para administrar la base de datos.

## **4.1. Definición de requerimientos**

Analizando los requerimientos y los objetivos buscados del Administrador de información, podemos determinar que las interfaces de usuario, la operatividad con la base de datos y los elementos propios de cada módulo de la aplicación son muy parecidos entre sí, lo cual ocasiona que el código fuente entre uno u otro módulo sea muy similar en su estructura y sintaxis.

Para el desarrollo del generador de código, a través del cual se implementará el Administrador de Información, debemos definir los alcances y limitaciones de la herramienta, así como plantear una propuesta de solución para su desarrollo, basándonos en la idea de la creación de las partes básicas necesarias para el funcionamiento del Sistema deseado, que pueda ser implementado bajo cualquier modelo de solución, es decir, que pueda trabajar en forma standalone, cliente-servidor o tres capas.

### **4.1.1. Características del generador de código**

El generador de código se encargará de analizar y describir las entidades existentes en la base de datos propuesta, para generar el código de las interfaces y el código de acceso a datos necesario para la administración de la información residente en la misma.

Los requerimientos con los que debe cumplir la herramienta de generación de código son:

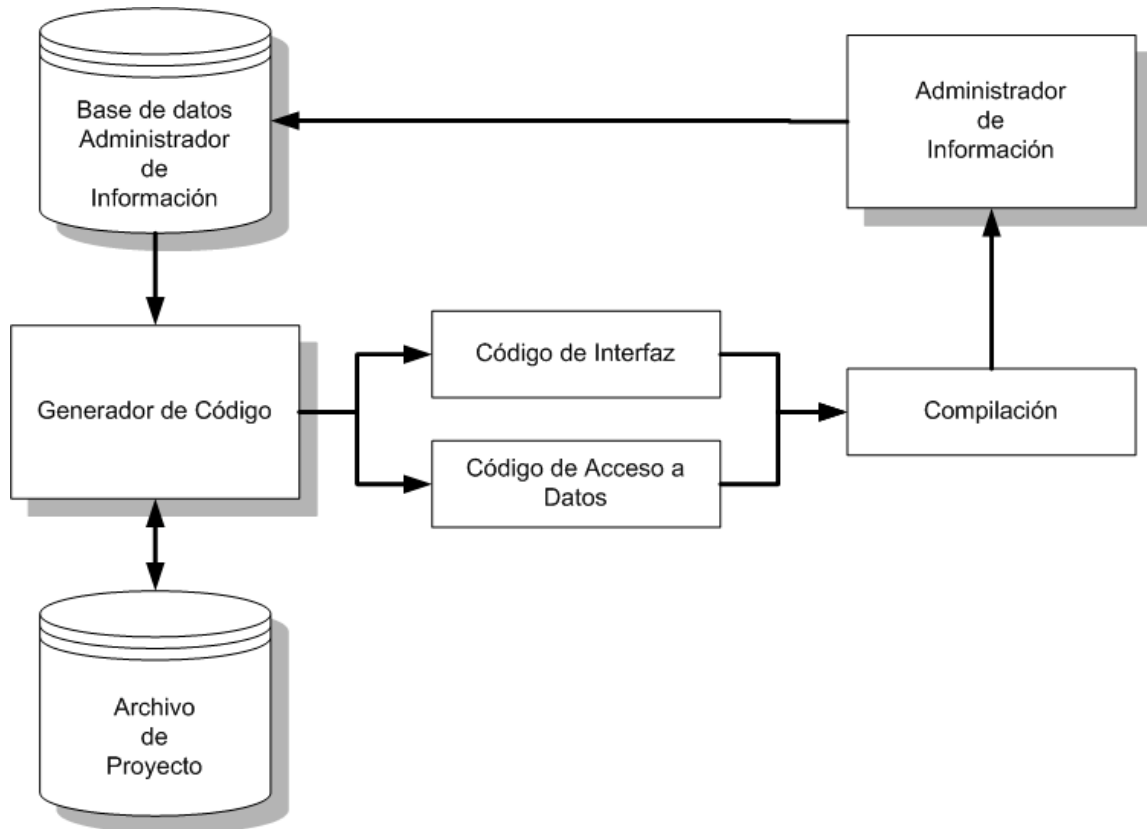
- El código generado por la herramienta debe cumplir, como mínimo, con los requerimientos definidos en el capítulo de análisis del Administrador de información.
- Contar con una interfaz de usuario sencilla.
- Buscar la interacción con las bases de datos relacionales normalizadas como Access y SQL Server.
- No considerar la generación de código de rutinas específicas que se refieran a las reglas del negocio.
- Generar rutinas para el manejo total de las tablas de la base de datos, es decir, alta, baja, cambio, consulta, así como rutinas para la administración de estas operaciones, de modo que cumplan siempre con la integridad de la información contenida en el modelo de la base de datos.
- Generar interfaces gráficas que expongan la operatividad entre el usuario y la información de la base de datos en su forma básica.
- Generar un modelo de seguridad que permita la integración de accesos al sistema en un esquema de Grupos-Permisos-Interfaz.
- Hacer un manejo controlado de las posibles excepciones durante la operatividad del sistema generado.
- Permitir la configuración del sistema deseado, desde su validación hasta su interfaz gráfica.
- Permitir que el sistema pueda trabajar en distintos esquemas de modelos de solución.

A comparación de los generadores de programas que existen en la actualidad, el generador de código a desarrollar más que apoyarse en la creación de programas en un lenguaje de tercera generación, se apoya en la definición de una base de datos relacional predefinida.

Basados en los puntos anteriores planteamos la funcionalidad con la que contará el generador de código.

## 4.2. Funcionalidad

El diagrama funcional de la herramienta generadora de código se muestra en la figura 4.1.



*Figura 4.1. Diagrama General del Proceso.*

En el diagrama de la figura se muestran dos bases de datos, la base de datos del Administrador de Información y el archivo de proyecto, este ultimo es una base de datos en formato de Access que almacena la configuración de la aplicación deseada.

El generador de código analiza las entidades de la base de datos del Administrador de Información. Con base en este análisis y a la configuración previa, realizada en el archivo de proyecto, genera el código de la Interfaz y de los componentes de acceso a los datos por separado, para controlar de una manera ordenada la información de la base de datos.

Después de compilarse el código generado, se tendrán los componentes necesarios para establecer el modelo de solución elegido por el usuario, que será capaz

de administrar la base de datos que se usó para generar el código del Administrador de Información.

La mayoría de las aplicaciones que tratan con datos, una vez que han conectado con el origen, solicitan directamente la información que precisan o ejecutan los comandos pertinentes, por lo cual necesitan conocer la estructura de la base de datos, ya que ésta, habitualmente, está disponible cuando el equipo de desarrollo planifica la aplicación.

Para el caso del generador de código es necesario conocer y analizar la estructura de bases de datos no definidas dentro del generador que cumpla con un modelo entidad relación, utilizando para ello los servicios del Proveedor de datos ADO.NET.

La mayoría de los orígenes de datos almacenan información sobre la estructura de los datos. Todos los RDBMS y las bases de datos locales utilizan esta información de esquema para saber cuáles son las tablas existentes, con qué columnas cuentan, qué tipos de datos corresponden a cada una de ellas, etc.

Si conectamos con un origen de datos cuya estructura desconocemos, la recuperación de la información de esquema, también conocida como *meta-información*, nos permitirá adaptar dinámicamente el funcionamiento del programa para que trate los datos apropiadamente.

Existen básicamente dos métodos para recuperar la información de esquema de la base de datos: utilizar los servicios del controlador OLE DB (*Object Linking & Embedding Database*), que es una tecnología para la creación y uso de distintos controladores de acceso a distintos orígenes de datos; o hacer uso de las instrucciones que el propio origen de datos facilite para obtener esa información.

El generador se basa principalmente en el primer método pues esto le da la facilidad a la herramienta de poder trabajar con otros orígenes de datos sin cambiar la estructura de recuperación de la información del esquema.

El proveedor OLE DB de ADO.NET no accede directamente a un origen de datos, sino que utiliza a modo de intermediario un controlador OLE DB existente. Los controladores OLE DB, como los proveedores ADO.NET, se ajustan a un estándar definido por Microsoft, contando todos ellos con métodos que permiten recuperar información de esquema del origen al que han conectado.

El generador de código internamente se compone de cuatro módulos principales, que se encuentran relacionados como se muestra en la figura 4.2, y que son:

- Módulo Nuevo / Abrir
- Módulo de Conexión a Datos

- Módulo de Configuración
- Módulo de Generación de Código

Estos módulos albergan la funcionalidad de la herramienta en donde la interfaz de usuario del generador de código está compuesta principalmente por un Módulo de Configuración, desde el cual se puede acceder directamente a todos los módulos que conforman la totalidad de la funcionalidad de la aplicación.

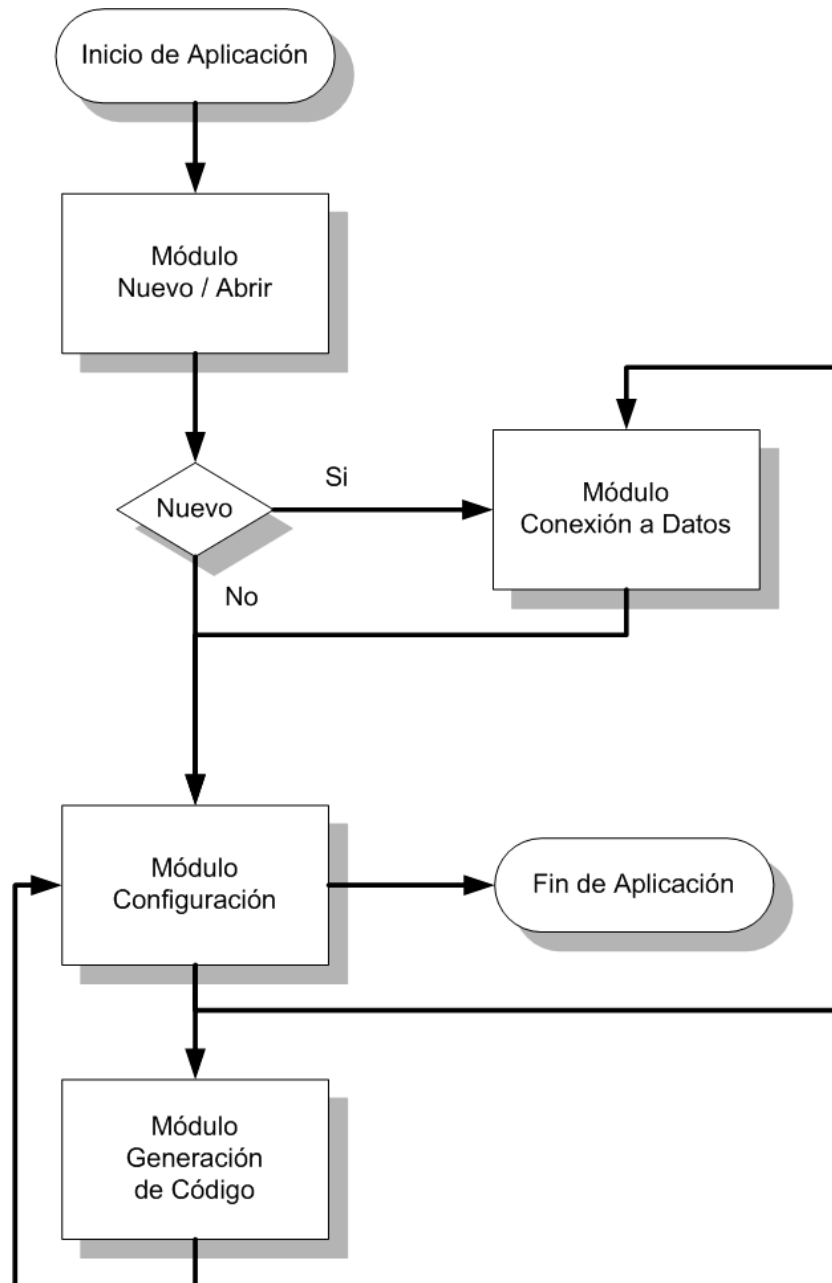


Figura 4.2. Diagrama de flujo del Generador de Código.



A excepción del Módulo de Generación de Código, cada uno de los módulos que conforman la herramienta poseen internamente estructuras de código que se encargan de la organización, configuración y creación de elementos de agrupación lógica (Colecciones).

En los siguientes subtemas de este capítulo detallamos la estructura de la interfaz de usuario y el funcionamiento lógico de cada uno de los módulos mostrados en el diagrama 4.2.

### 4.2.1. Módulo Nuevo / Abrir

Este módulo se divide en dos partes; una que permite la creación de un nuevo archivo de proyecto para guardar la información de configuración de la aplicación que se desea crear, y otra que se encarga de la carga de un archivo de proyecto ya existente en el generador.

El diagrama entidad-relación de la figura 4.3, nos muestra siete entidades que conforman el formato que mantienen los archivos de proyecto de configuración del generador de código.

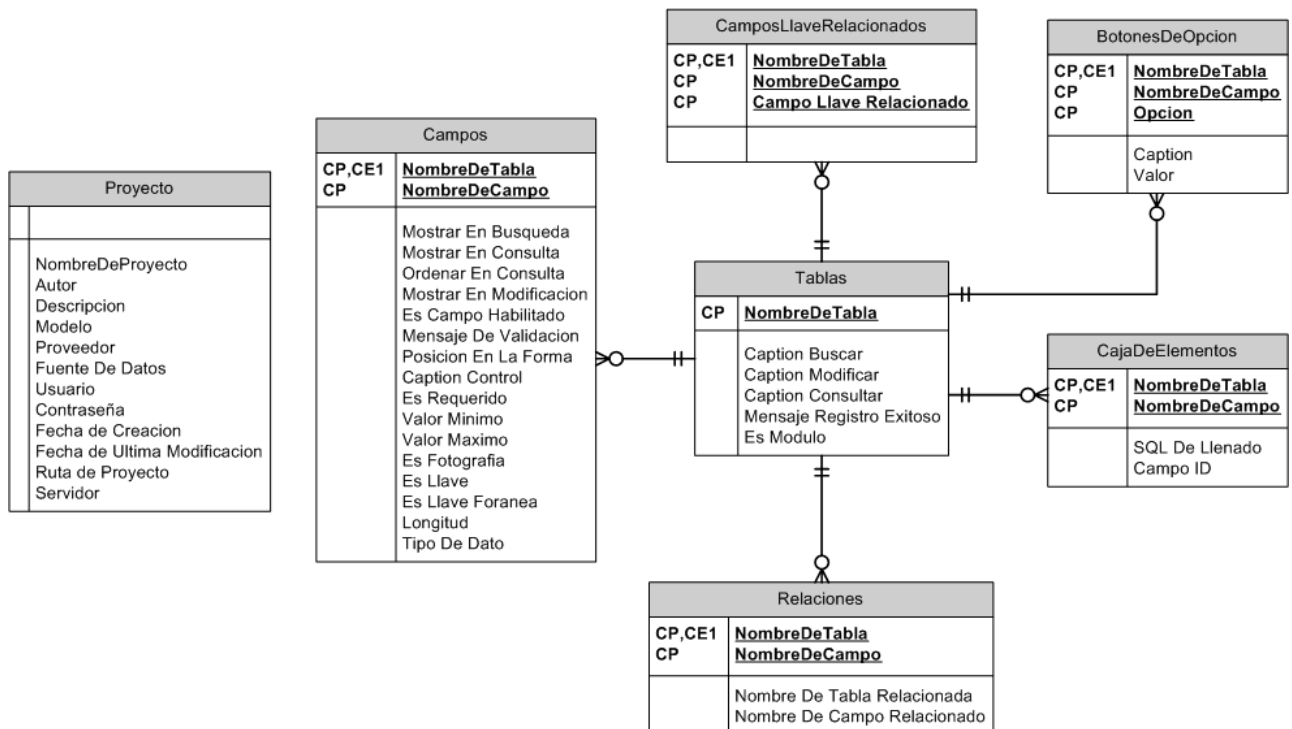


Figura 4.3. Modelo de datos del archivo de proyecto.

La estructura de estas entidades se encarga de almacenar el esquema del modelo de datos de una fuente de información, así como, la configuración que el generador de código requiere del mismo. Las características principales de cada una de estas entidades son:

*Proyecto*

Esta entidad almacena la información referente al archivo de proyecto y a la fuente de datos y sus columnas se describen en la tabla 4.1.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeProyecto</b>	VARCHAR(255)	Sí	Nombre con el que se describe al proyecto
<b>Autor</b>	VARCHAR(255)	Sí	Nombre del autor del proyecto
<b>Descripcion</b>	LONGCHAR	Sí	Descripción del proyecto
<b>Modelo</b>	INTEGER	Sí	Modelo de datos en el que se generará el proyecto
<b>Proveedor</b>	VARCHAR(255)	Sí	Nombre del Proveedor de datos que se utilizará para la conexión con la fuente de información.
<b>Fuente De Datos</b>	VARCHAR(255)	Sí	Ruta de ubicación física de la fuente de datos que se utilizara para la conexión con la fuente de información.
<b>Usuario</b>	VARCHAR(255)	Sí	Nombre del usuario para la autenticación de la conexión con la base de datos
<b>Contraseña</b>	VARCHAR(255)	Sí	Clave de acceso para la autenticación de la conexión con la base de datos

*Tabla 4.1. Columnas de entidad Proyecto. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
<b>Fecha de Creacion</b>	DATETIME	Sí	Fecha en que fue creado el archivo de proyecto
<b>Fecha de Ultima Modificacion</b>	DATETIME	Sí	Fecha en que fue efectuada la última modificación del archivo de proyecto
<b>Ruta de Proyecto</b>	VARCHAR(255)	Sí	Ruta de ubicación física en la que se encuentra el archivo de proyecto
<b>Servidor</b>	VARCHAR(255)	Sí	Nombre del Servidor de datos que se utilizará para la conexión con la fuente de información.

*Tabla 4.1. Columnas de entidad Proyecto.*

### Tablas

Esta entidad se encarga de almacenar la información del esquema de las tablas de la fuente de datos, para la configuración de elementos contenedores de objetos gráficos, sus columnas se describen en la tabla 4.2.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeTabla</b>	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad
<b>Caption Buscar</b>	VARCHAR(255)	Sí	Texto de captura que aparecerá como mensaje en el encabezado del contenedor de objetos gráficos Buscar

*Tabla 4.2. Columnas de entidad Tablas. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
<b>Caption Modificar</b>	VARCHAR(255)	Sí	Texto de captura que aparecerá como mensaje en el encabezado del contenedor de objetos gráficos Modificar
<b>Caption Consultar</b>	VARCHAR(255)	Sí	Texto de captura que aparecerá como mensaje en el encabezado del contenedor de objetos gráficos Consultar
<b>Mensaje Registro Exitoso</b>	LONGCHAR	Sí	Texto del mensaje que aparecerá como mensaje de diálogo al modificarse o agregarse correctamente un registro mediante el contenedor de objetos gráficos Modificar
<b>Es Modulo</b>	BIT	Sí	Este campo es para identificar a la entidad del modelo de datos como un Módulo o catálogo de la aplicación que se va a generar

*Tabla 4.2. Columnas de entidad Tablas.*

### Relaciones

Esta entidad almacena el esquema de relaciones entre entidades de la fuente de información del sistema que se desea generar, sus columnas se describen en la tabla 4.3.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeTabla</b>	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad

*Tabla 4.3. Columnas de entidad Relaciones. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeCampo</b>	VARCHAR(255)	No	Nombre del campo existente en la entidad tabla que sirve de identificador único de la entidad
<b>Nombre De Tabla Relacionada</b>	VARCHAR(255)	Sí	Nombre de la tabla con la que se relaciona la entidad
<b>Nombre De Campo Relacionado</b>	VARCHAR(255)	Sí	Nombre del campo de relación mediante el cual se consigue la relación de las entidades

*Tabla 4.3. Columnas de entidad Relaciones.*

#### *CamposLlaveRelacionados*

Esta entidad sirve para relacionar una entidad con una campo llave de otra entidad dentro del modelo de datos, sus columnas se describen en la tabla 4.4.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeTabla</b>	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad
<b>NombreDeCampo</b>	VARCHAR(255)	No	Nombre del campo existente en la entidad tabla que sirve de identificador único de la entidad
<b>Campo Llave Relacionado</b>	VARCHAR(255)	No	Nombre del campo llave que relaciona a la entidad

*Tabla 4.4. Columnas de entidad CamposLlaveRelacionados.*

### Campos

Esta entidad almacena la información del esquema de los campos de las tablas de la fuente de datos para la configuración de los objetos gráficos de la interfaz de usuario, sus columnas se describen en la tabla 4.5.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeTabla</b>	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad
<b>NombreDeCampo</b>	VARCHAR(255)	No	Nombre del campo existente en la entidad tabla que sirve de identificador único de la entidad
<b>Mostrar En Búsqueda</b>	BIT	Sí	Campo lógico que permite determinar la creación de un objeto gráfico sobre el contenedor de objetos gráficos Buscar
<b>Mostrar En Consulta</b>	BIT	Sí	Campo lógico que permite determinar la creación de un objeto gráfico sobre el contenedor de objetos gráficos Consultar
<b>Ordenar En Consulta</b>	BIT	Sí	Campo lógico que permite la ordenación sobre el contenedor de objetos gráficos Consultar
<b>Mostrar En Modificación</b>	BIT	Sí	Campo lógico que permite determinar la creación de un objeto gráfico sobre el contenedor de objetos gráficos Modificar
<b>Es Campo Habilitado</b>	BIT	Sí	Campo lógico que identifica a un campo como campo de borrado lógico

*Tabla 4.5. Columnas de entidad Campos. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
<b>Mensaje De Validacion</b>	LONGCHAR	Sí	Texto del mensaje que aparecerá como mensaje de diálogo, al tratar de ingresar un valor incorrecto en algún campo de la entidad
<b>Posicion En La Forma</b>	INTEGER	Sí	Valor numérico que proporciona la ubicación que ocupará el objeto gráfico dentro del contenedor de objetos gráficos
<b>Caption Control</b>	VARCHAR(255)	Sí	Texto de captura que describe el valor del dato que representa el campo en la aplicación
<b>Es Requerido</b>	BIT	Sí	Campo lógico que indica que el campo no acepta valores nulos
<b>Valor Minimo</b>	INTEGER	Sí	Valor numérico del límite inferior que puede aceptar el campo
<b>Valor Maximo</b>	INTEGER	Sí	Valor numérico del límite superior que puede aceptar el campo
<b>Es Fotografia</b>	BIT	Sí	Campo lógico que indica si el campo va a guardar la información de un objeto grafico

*Tabla 4.5. Columnas de entidad Campos. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
Es Llave	BIT	Sí	Campo lógico que indica si el campo es parte de la llave de la entidad
Es Llave Foranea	BIT	Sí	Campo lógico que indica si el campo es parte de la llave de la entidad y representa una liga con otra entidad del modelo de datos
Longitud	INTEGER	Sí	Valor numérico que establece el número de caracteres máximo que puede contener el campo
Tipo De Dato	VARCHAR(50)	Sí	Nombre del tipo de dato que representa el campo según el esquema de valores de tipos permitidos por .NET

Tabla 4.5. Columnas de entidad Campos.

#### BotonesDeOpcion

Esta entidad almacena la información de los posibles valores definidos que puede almacenar y representar en la interfaz de usuario un campo, sus columnas se describen en la tabla 4.6.

Columnas	Tipo de datos	Nulo	Descripción
NombreDeTabla	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad

Tabla 4.6. Columnas de entidad BotonesDeOpcion. (Continúa)



Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeCampo</b>	VARCHAR(255)	No	Nombre del campo existente en la entidad tabla que sirve de identificador único de la entidad
<b>Opcion</b>	INTEGER	No	Valor que numera la cantidad de opciones con que cuenta el campo
<b>Caption</b>	VARCHAR(255)	Sí	Texto de captura que describe el valor del dato que representa el campo
<b>Valor</b>	VARCHAR(255)	Sí	Valor de cualquier tipo que representa el valor que se guardará en la fuente de datos, respecto a la opción seleccionada por el usuario

*Tabla 4.6. Columnas de entidad BotonesDeOpcion.*

### *CajaDeElementos*

Esta entidad almacena la información de las cadenas de consulta para la recuperación de los posibles valores que puede almacenar y representar en la interfaz de usuario un campo, sus columnas se describen en la tabla 4.7.

Columnas	Tipo de datos	Nulo	Descripción
<b>NombreDeTabla</b>	VARCHAR(255)	No	Nombre de la tabla que sirve de identificador único de la entidad

*Tabla 4.7. Columnas de entidad BotonesDeOpcion. (Continúa)*

Columnas	Tipo de datos	Nulo	Descripción
NombreDeCampo	VARCHAR(255)	No	Nombre del campo existente en la entidad tabla que sirve de identificador único de la entidad
SQL De Llenado	LONGCHAR	Sí	Cadena de consulta en formato SQL para el llenado automático de los objetos STComboBox
Campo ID	VARCHAR(255)	Sí	Nombre del campo ID que representa el valor de identificación dentro de los objetos STComboBox

Tabla 4.7. Columnas de entidad BotonesDeOpcion.

Las entidades definidas anteriormente describen un archivo de proyecto que es manipulado mediante las interfaces presentadas por el Módulo Nuevo / Abrir del generador de código.

El Módulo Nuevo / Abrir presenta al usuario, al elegir la creación de un nuevo proyecto, la interfaz mostrada en la figura 4.4, en donde el usuario debe seleccionar entre tres modelos disponibles, el modelo de solución en el que desea implementar la aplicación, que son: Standalone, Cliente-Servidor y Tres capas. El modelo de solución especificado determina la forma en que el generador de código creará y organizará los archivos de código fuente, así como las opciones específicas de compilación que contendrá el proyecto para su correcto funcionamiento.

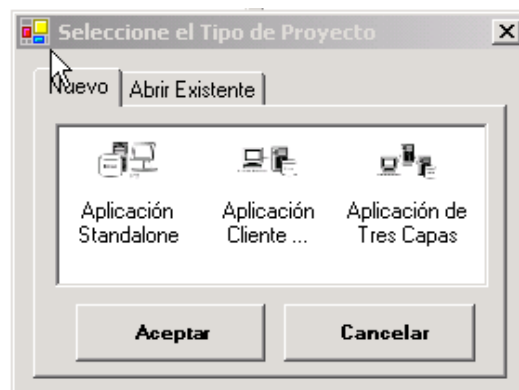


Figura 4.4. Nuevo Proyecto.

La Interfaz que se presenta al seleccionar abrir un proyecto existente (Figura 4.5.), permite la búsqueda de un archivo de proyecto con extensión .pry. Esta selección se realiza para continuar con la configuración del mismo o para la creación de la aplicación basada en la información contenida en el mismo.

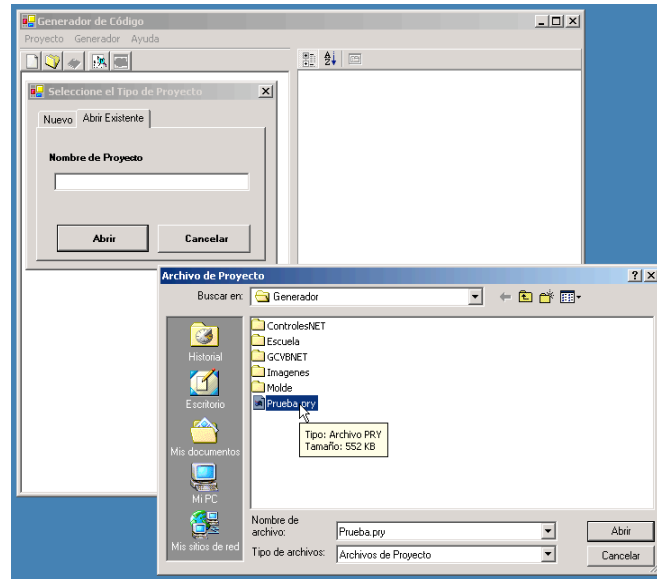


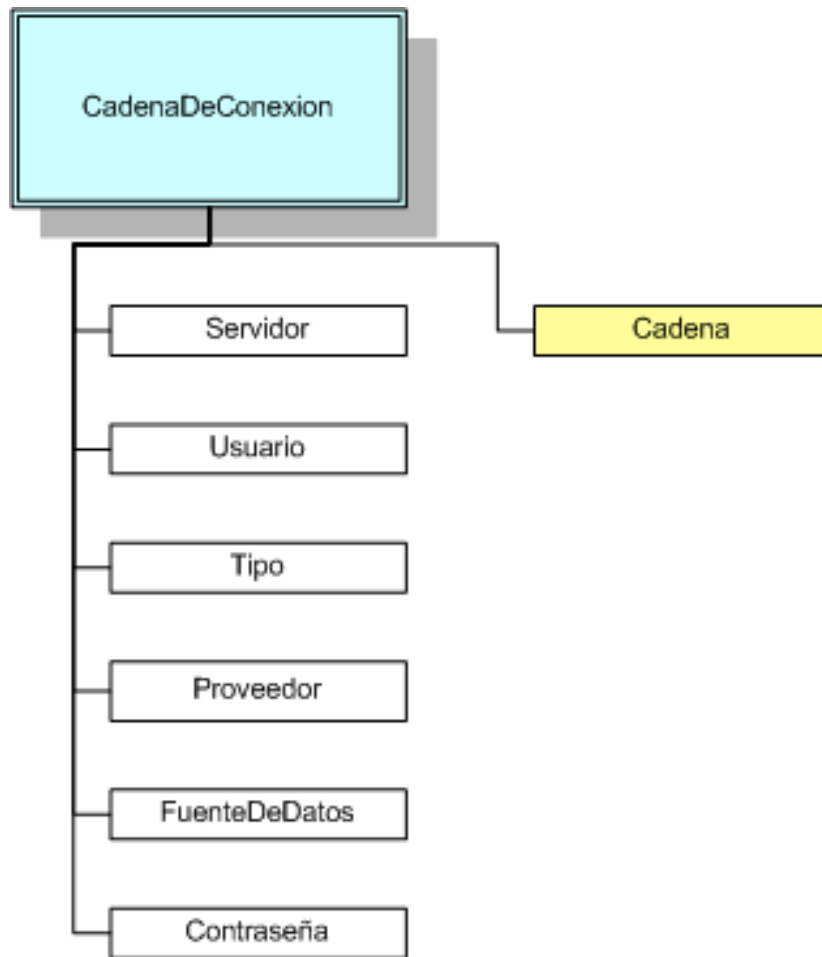
Figura 4.5. Abrir un proyecto existente.

#### 4.2.2. Módulo de Conexión a Datos

El módulo de conexión a datos se encarga de la creación y comprobación de las cadenas de conexión requeridas por el controlador OLEDB, para la recuperación de los esquemas de base de datos de la fuente de información desde la que se quiere crear la aplicación. Los proveedores de datos con los que puede funcionar la aplicación son SQL Server y Access.

La cadena de conexión a datos que crea el módulo se basa en la estructura del objeto *CadenaDeConexion* mostrado en la figura 4.6.

El objeto *CadenaDeConexion* presenta seis propiedades y un método; las propiedades almacenan la información de los parámetros de conexión necesarios por el controlador OLEDB para acceder a la fuente de datos, mientras que el método *Cadena* se encarga del acomodo y armado de estas propiedades para la creación de los parámetros de la conexión.

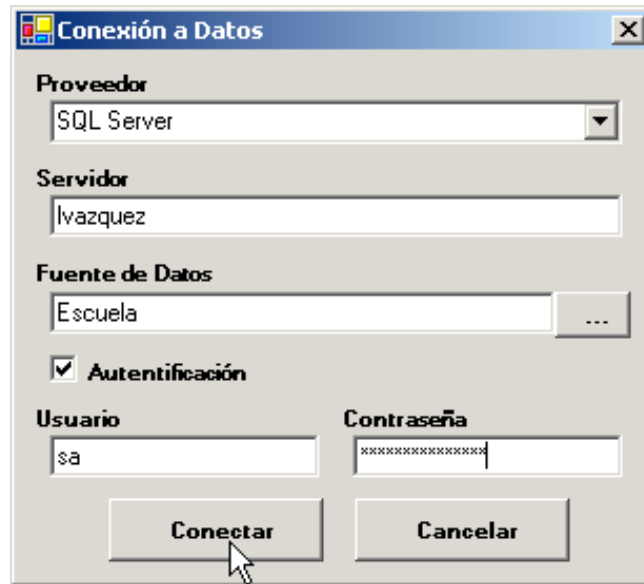


*Figura 4.6. Objeto CadenaDeConexion.*

Entre estas propiedades tenemos: el nombre del servidor en donde se encuentra la base de datos, el usuario y la contraseña para su autenticación en la fuente de información, así como el nombre del controlador específico que se encargará de proveer los datos.

La interfaz que presenta este módulo para la asignación de valores a las propiedades del objeto *CadenaDeConexion* es la mostrada en la figura 4.7.

En esta interfaz se define el tipo de proveedor de datos, el servidor y la base de datos con la que se hará la conexión, así como el usuario y contraseña requeridos para la autenticación, en caso de que la fuente de información lo requiera.



*Figura 4.7. Módulo de conexión a datos.*

### **4.2.3. Módulo de Configuración**

Este módulo Inicialmente recupera la información de la fuente de datos proporcionada por el archivo de proyecto que el usuario eligió, o por la conexión a datos que creó y validó el módulo de conexión a datos.

Posteriormente analiza los datos y se encarga del llenado de los modelos de clase, definiendo algunos valores para las propiedades de los objetos que por la naturaleza de la entidad o los valores permitidos por el tipo de datos que está posee son requeridos.

Las estructuras de los modelos de clase, definidos para la generación de código, se encargan de la descripción del esquema de datos de la aplicación que se desea generar.

Es necesario el llenado de las estructuras de los modelos de clase para contener las entidades, relaciones y campos de los esquemas de modelos de datos de la fuente de información y de la configuración de la aplicación deseada. Las estructuras de estos modelos se basan en colecciones de objetos jerarquizados, como se muestra en la figura 4.8.

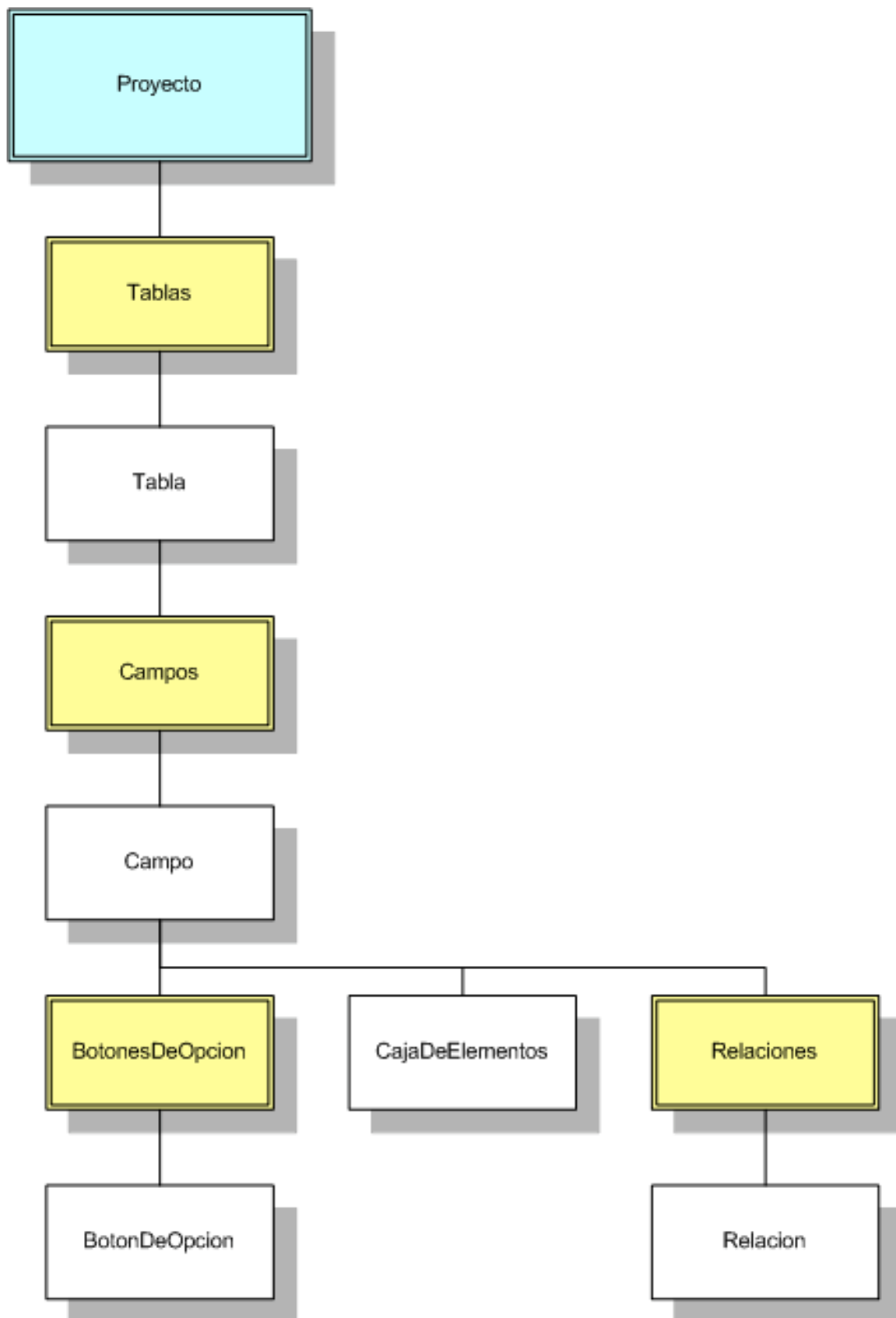


Figura 4.8. Jerarquía de clases.

Dentro de esta estructura el objeto principal en el que se basa todo el modelo, es el objeto *Proyecto*, que agrupa colecciones de objetos *Tablas*, *Campos*, *BotonesDeOpcion* y *Relaciones*, así como elementos de clase como son *Tabla*, *Campo*, *BotonDeOpcion*, *CajaDeElementos* y *Relacion*.

Estos objetos definen propiedades y métodos que podemos describir más a detalle de la siguiente manera.

### *Proyecto*

El objeto proyecto que se muestra en la figura 4.9, contiene las propiedades de configuración y creación del código fuente, así como los métodos para guardar y recuperar las estructuras de datos diseñadas en los archivos de proyecto.

Entre las propiedades de este objeto encontramos tres grupos de elementos de información distintos que son:

- Información del proyecto. *FechaDeUltimaModificacion*, *FechaDeCreacion*, *Nombre*, *Descripcion*, *Autor* y *RutaDeProyecto*.
- Información de conexión a datos. *Proveedor*, *FuenteDeDatos*, *Usuario*, *Contraseña* y *Servidor*.
- Información del Modelo de Solución. La propiedad *Modelo* establece el modelo de solución en el que se desarrollara la aplicación.

### *Tablas*

El objeto *Proyecto* a su vez agrupa una colección de objetos llamada *Tablas*, esta colección almacena objetos *Tabla* (figura 4.10), que contienen todas las propiedades de configuración de las entidades contenidas en el esquema de base de datos.

Las propiedades de este objeto almacenan la definición de la configuración que presentarán las interfaces de la aplicación deseada y definirán la construcción y forma del código para las distintas entidades del modelo de datos.

Entre las propiedades de este objeto existe la propiedad *EsModulo*, que define el comportamiento de la entidad en la aplicación final, un valor Verdadero en esta propiedad creara el código de un modulo a partir de esta entidad, mientras que un valor Falso creara el código para la administración de un catálogo.

El objeto provee un método llamado *CargarTablas* para la lectura de todas las colecciones de objetos entidades de la misma.

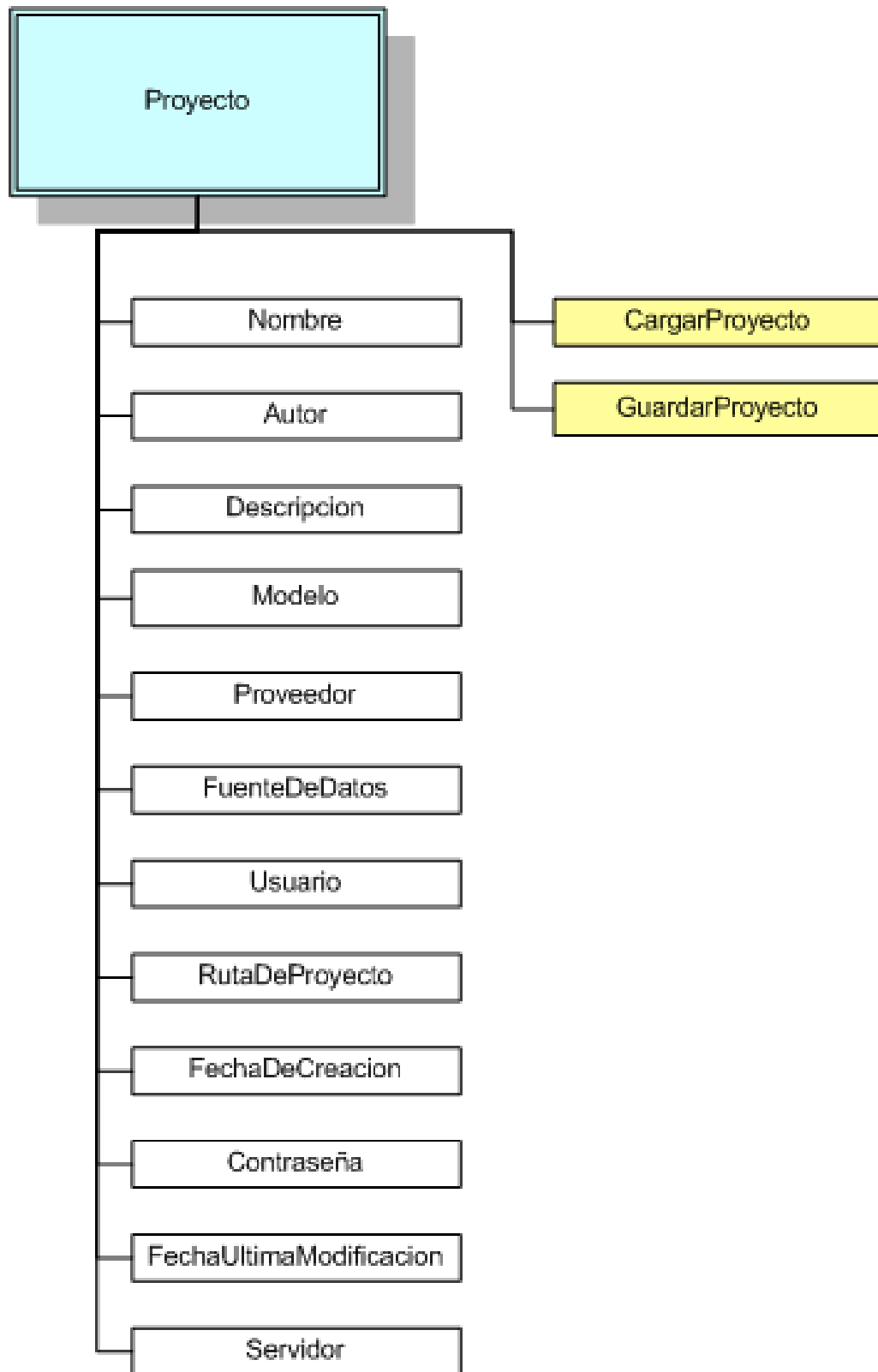


Figura 4.9. Estructura de objeto Proyecto.



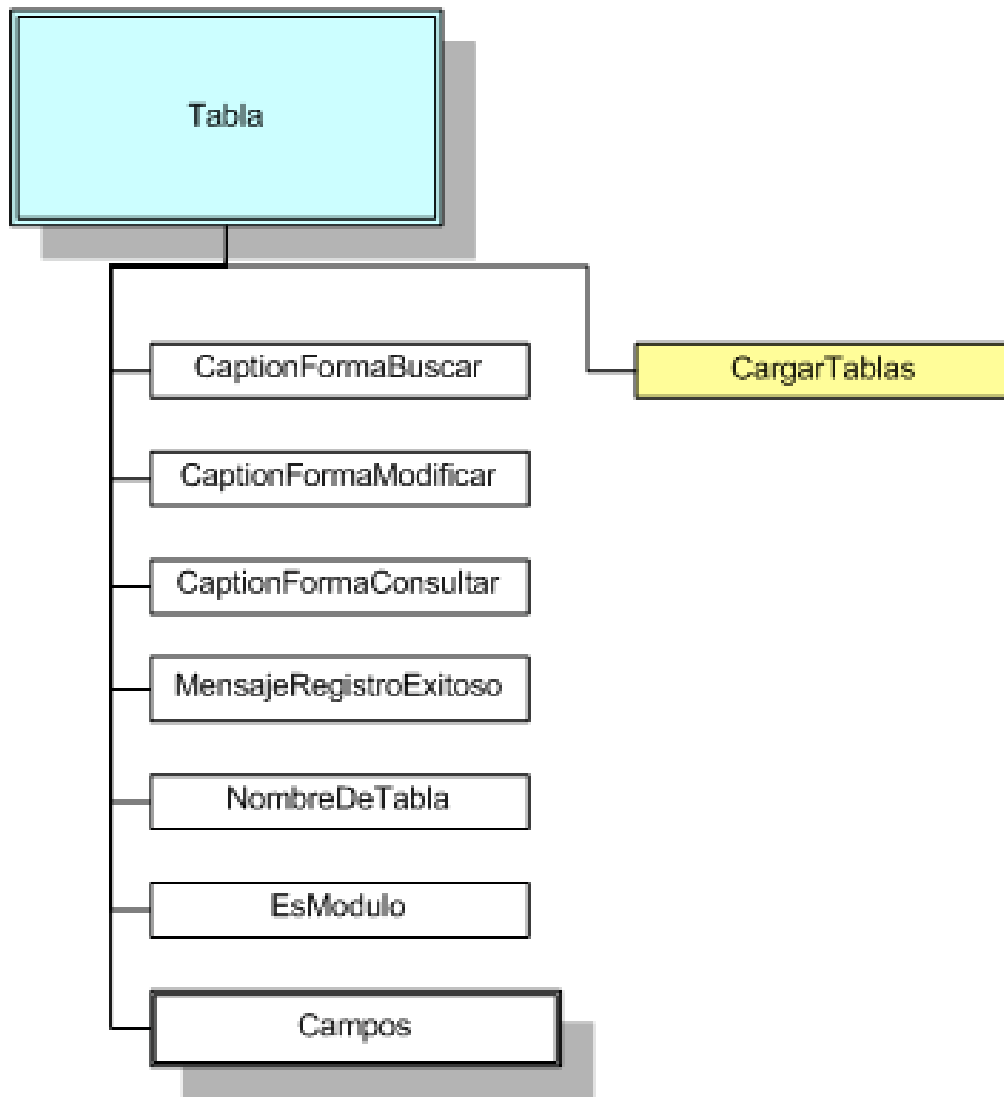


Figura 4.10. Estructura de objeto Tablas.

A su vez, el objeto *Tabla* agrupa en una de sus propiedades una colección de objetos *Campos* que al igual que la colección *Tablas* almacena la información correspondiente al esquema de datos de los campos en forma individual, en un objeto *Campo* como se muestra en la figura 4.11.

Este objeto provee un método llamado *CargaCampos*, el cual se encarga de la lectura de todos los campos contenidos en una colección de objetos entidades del esquema de base de datos proporcionado.

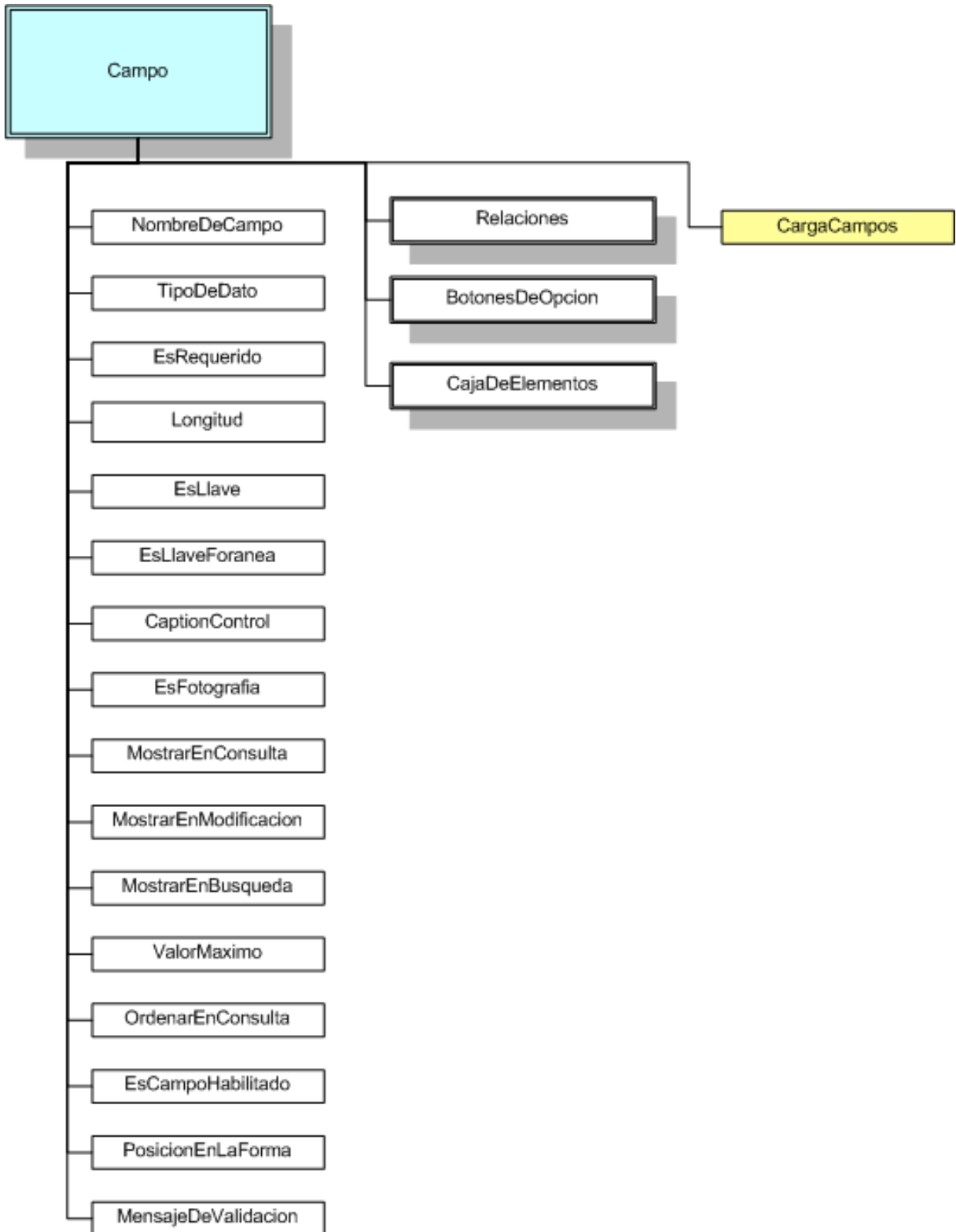


Figura 4.11. Estructura de objeto Campo.

El método *CargaCampos* define el valor de algunas propiedades que por la naturaleza de los campos de las entidades son requeridos como son: *TipoDeDato*, *EsRequerido*, *Longitud*, *EsLlave*, *EsLlaveForanea*, *ValorMaximo*.

Este objeto contiene dos colecciones *Relaciones* y *BotonesDeOpcion* y un objeto *CajaDeElementos*.

La colección *Relaciones* contiene objetos *Relacion* (figura 4.12) y se encarga de almacenar las propiedades de configuración, para la creación de restricciones, y el enlazamiento dinámico de información entre las entidades del esquema de base de datos.

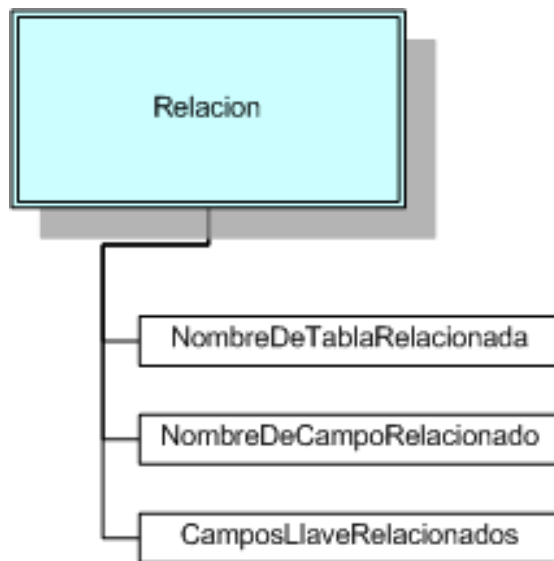


Figura 4.12. Estructura de objeto *Relacion*.

El objeto *Relacion* posee tres propiedades que almacenan la información de la relación existente entre grupos de campos de entidad que son: *NombreDeTablaRelacionada*, que es el nombre de la tabla que contiene los campos con los que se relaciona el campo de la entidad; *NombreDeCampoRelacionado*, que es el nombre del campo de la entidad y *CamposLlaveRelacionados*, que es un arreglo de nombres de campos que sirven de liga para el campo de entidad en otra tabla.

La colección *BotonesDeOpcion* contiene objetos *BotonDeOpcion* (figura 4.13), que guardan la estructura que deberán poseer los elementos gráficos de la interfaz de usuario para poder almacenar un número de elementos definidos de valores posibles que puede tomar un campo.

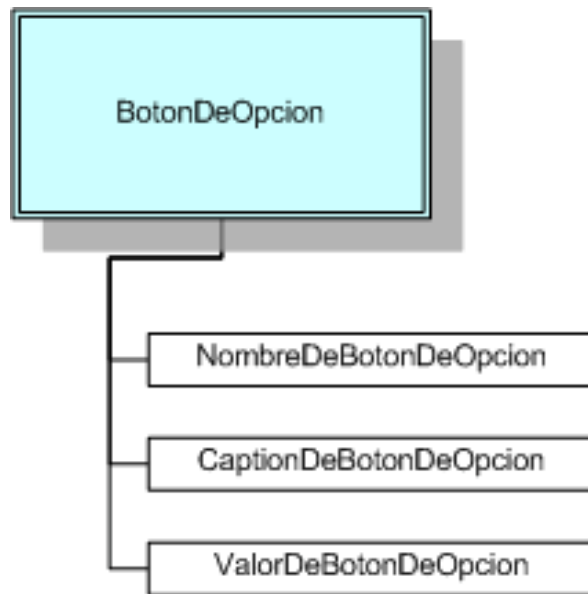


Figura 4.13. Estructura de objeto *BotonDeOpcion*.

El objeto *CajaDeElementos* (figura 4.14), se encarga de crear una estructura para el almacenamiento y despliegue dinámico de elementos de un número indefinido de valores posibles que puede tomar un campo.

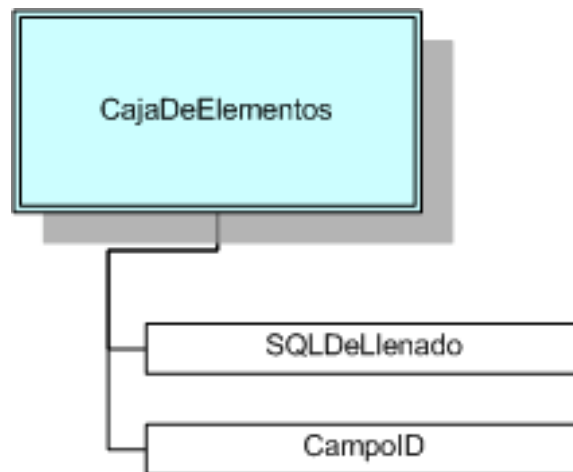


Figura 4.14. Estructura de objeto *CajaDeElementos*.

La propiedad *SQLDeLlenado* especifica una cadena con estructura en lenguaje SQL, para la lectura de datos desde otras entidades que sirvan de referencia para el despliegue de información en la interfaz de usuario.

Para la configuración de cada uno de los modelos de objetos definidos anteriormente, se hace uso de un módulo de configuración, el cual se compone de una interfaz gráfica, que permite la configuración de cada uno de los elementos del esquema de datos de la fuente de información.

La interfaz del módulo de configuración se divide en 4 zonas principales, que conforman la funcionalidad de la herramienta, como se muestra en la figura 4.15.

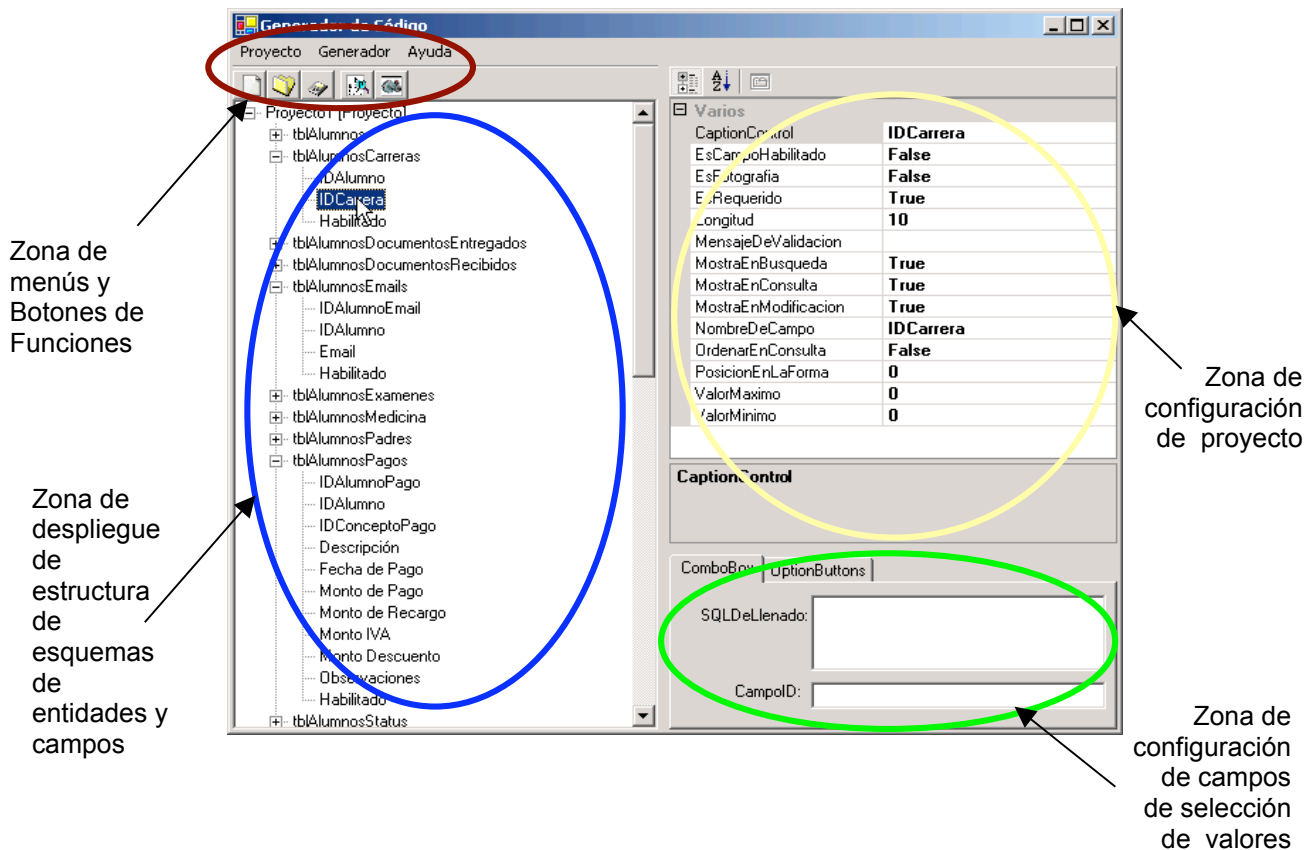


Figura 4.15. Módulo de Configuración.

### Zona de menús

En la zona de menús y botones de opción se presentan las funciones disponibles a las que se tiene acceso, dependiendo de la información con que se cuenta en un determinado momento.

### Zona de despliegue de estructura de esquemas

En la zona de despliegue de estructura de esquemas de entidades y campos se muestra en forma de árbol el esquema de la base de datos que se seleccionó, en el

módulo de nuevo/abrir proyecto para la selección y configuración de los elementos de la fuente de información.

*Zona de configuración de proyecto*

En la zona de configuración de proyecto se presenta un panel que muestra las posibles opciones que se pueden hacer sobre el elemento correspondiente del modelo de árbol mostrado en la zona de despliegue. Esta zona puede cambiar de forma dinámica; dependiendo del tipo de elemento seleccionado, estos elementos pueden ser:

- Proyecto. En este caso el panel presentado muestra las propiedades de configuración del proyecto que se está configurando, como se muestra en la figura 4.16.

Varios	
Autor	Luis Antonio Vázquez Ramos
Contraseña	sa
Descripcion	Administrador de Información de Escuela
FechaDeCreacion	13/06/2003 04:34 p.m.
FechaDeUltimaModificacion	13/06/2003 04:34 p.m.
FuenteDeDatos	Initial Catalog=Escuela;
Modelo	TresCapas
Nombre	Proyecto1
Proveedor	Provider=SQLOLEDB.1;
RutaDeProyecto	C:\Proyecto1.pry
Servidor	DBS
Usuario	Integrated Security=SSPI;Persist Security Info=False;

*Figura 4.16. Panel de configuración de proyecto.*

- Tabla. Como se muestra en la figura 4.17, en este caso el panel de configuración muestra las propiedades de la entidad tabla, mediante la cual se crean los módulos que permitirán la funcionalidad de la aplicación.

Varios	
CaptionFormaBuscar	Busqueda de Alumnos
CaptionFormaConsultar	Consulta de Alumnos
CaptionFormaModificar	Modificación de Alumnos
EsModulo	True
MensajeRegistroExitoso	El alumno fue registrado exitosamente.
NombreDeTabla	tblAlumnos

*Figura 4.17. Panel de configuración de tablas.*

- Campo. Para el caso de la configuración de los elementos campo de las tablas, el panel de configuración es el mostrado en la figura 4.18.

Varios	
CaptionControl	Edad
EsCampoHabilitado	False
EsFotografia	False
EsRequerido	True
Longitud	0
MensajeDeValidacion	La edad debe ser de tipo numérico.
MostraEnBusqueda	False
MostraEnConsulta	False
MostraEnModificacion	True
NombreDeCampo	Edad
OrdenarEnConsulta	False
PosicionEnLaForma	0
ValorMaximo	100
ValorMinimo	0

Figura 4.18. Panel de configuración de campos.

#### Zona de configuración de campos de selección

La zona de configuración de campos de selección de valores se encarga de la configuración de los objetos de selección de grupos de valores posibles, permitiendo el llenado y la identificación de los elementos contenidos en cada objeto.

La selección de valores puede presentarse en dos formas, que dependen de la cantidad de valores posibles, y son:

- Forma de selección Estática. En este caso el número de valores está definido y es un número reducido de posibles opciones, como por ejemplo el Sexo, el estado civil, etc. (figura 4.19).

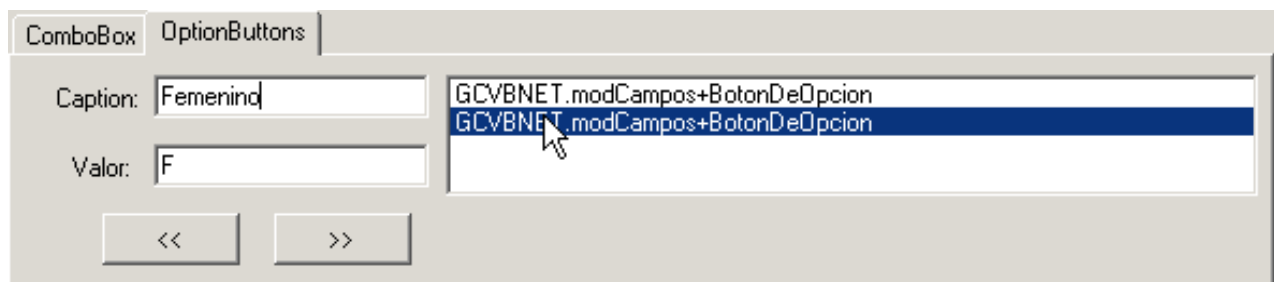


Figura 4.19. Forma de selección estática.

- Forma de selección Dinámica. En este caso el número de valores no está definido y puede crecer de forma dinámica, por lo cual se hace uso de una cadena de consulta SQL para el llenado de las posibles opciones (figura 4.20).



The screenshot shows a configuration window with two tabs: 'ComboBox' and 'OptionButtons'. The 'OptionButtons' tab is selected. Below the tabs, there are two input fields. The first is labeled 'SQLDeLlenado:' and contains the text 'Select Nombre & [Apellido Paterno] & [Apellido Materno] From tblAlumnos'. The second is labeled 'CampoID:' and contains the text 'IDAlumno'. A mouse cursor is pointing at the 'OptionButtons' tab.

*Figura 4.20. Forma de selección dinámica.*

#### **4.2.4. Módulo de Generación de Código**

Con la configuración previa del análisis de los esquemas de base de datos de la fuente de información y de la configuración de la aplicación deseada, basándose en los modelos de clase definidos por el módulo de configuración, este módulo se encarga de la creación del código fuente.

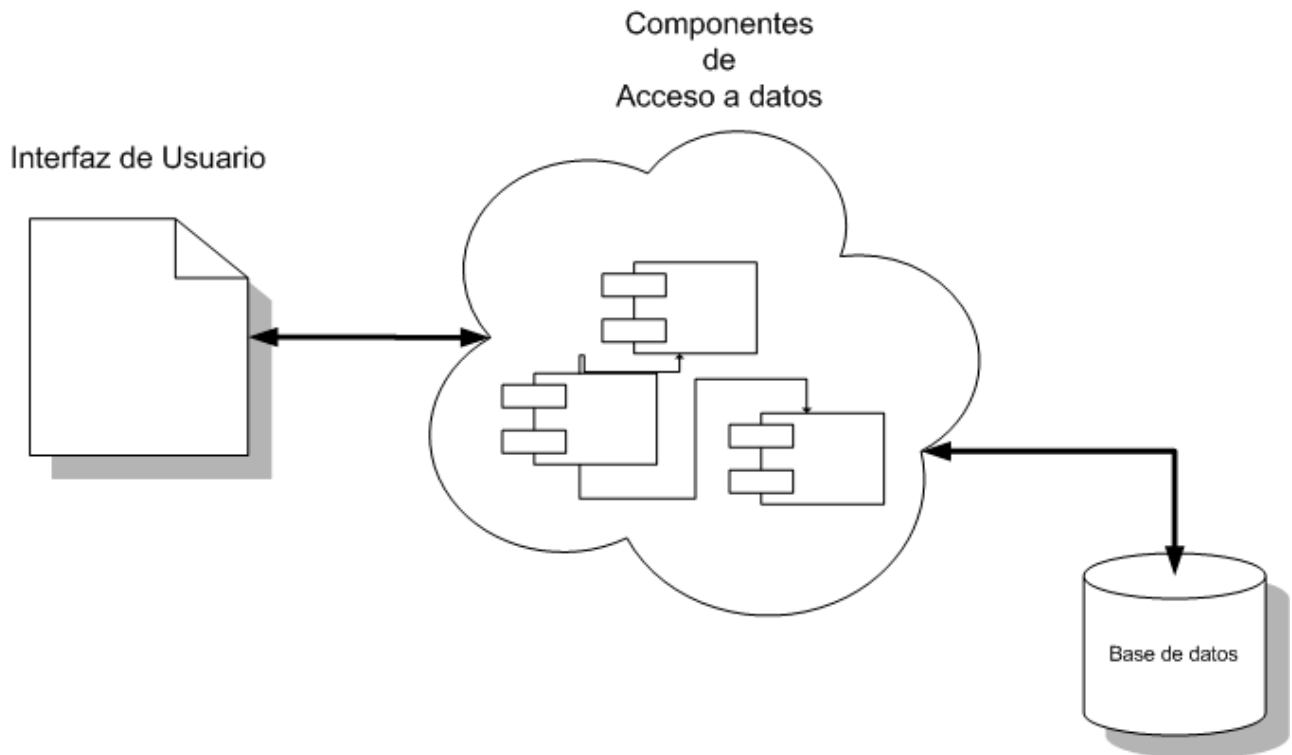
Para la creación del código fuente mencionado es necesario contar con un código base que sirva de plantilla para la implementación de las distintas interfaces de usuario, los objetos de validación y los componentes de acceso a la base de datos de cada una de las entidades contenidas en la fuente de información.

### **4.3. Generación del código base**

El código base para la creación de la aplicación debe de ser lo más optimo posible, permitiendo el cambio dinámico de variables y la creación de algoritmos de consulta y modificación de forma automática.

Es necesario definir como punto central la creación del código en dos partes básicas, que aplique para cualquier modelo de solución: la interfaz de usuario y los componentes de acceso a datos de la aplicación, como se muestra en la Figura 4.21.





*Figura 4.21. Elementos del modelo de solución.*

### **4.3.1. Interfaz de Usuario**

La interfaz de usuario es un panel de control físico con entidades incluidas, como son: botones, menús, desplegados de información, entre otros.

Las interfaces gráficas presentan al usuario un espacio donde puede visualizar su información, la cual es modificada a través de una acción directa. Esta manipulación debe ser cuidadosamente elaborada, ya que las interfaces de manipulación directa pueden ser tan complejas como para ser programadas y por tanto exigir mayores requerimientos del sistema en cuestión.

Comúnmente los usuarios juzgan un sistema por su interfaz, por lo que, para el diseño de interfaces de usuario, se deben tomar en cuenta las necesidades, experiencia y capacidades de los usuarios. El diseño de una interfaz de usuario pobre es la razón por la cual muchos sistemas nunca son usados.

Los usuarios deben involucrarse en el proceso de diseño y el diseño de las interfaces de usuario deben refinarse hacia rápidos prototipos.

Algunas de las características buscadas principalmente en una interfaz de usuario son:

- La interfaz debe basarse en los términos del usuario y conceptos informáticos simples.
- El sistema deberá mostrar un nivel apropiado de consistencia.
- Los elementos de la interfaz deben mantener un formato similar.
- El diseño de interfaces debe ser enfocada al usuario.
- El despliegue gráfico puede utilizarse para representar tendencias y valores aproximados.
- El despliegue digital debe presentarse sólo cuando es requerida la precisión.
- Si una función de la interfaz opera en una forma conocida, el usuario debe ser capaz de predecir la operación de una función similar.
- El sistema debe proveer alguna ayuda cada vez que el usuario cometa un error y dar la posibilidad a éste de corregir el error antes de ejecutarlo.

Para la creación del código base de las interfaces se utilizan algunos objetos gráficos propios del entorno de desarrollo .NET y se crean otros basados en la herencia de la programación orientada a objetos que posee el lenguaje de programación, estos objetos según su funcionalidad dentro de la interfaz pueden ser clasificados como se muestra en la figura 4.22.

Inicialmente toda Interfaz de Usuario proporciona un contenedor de objetos gráficos, que es llamado de diferentes maneras dependiendo de la plataforma de Sistema Operativo de que se trate.

Para el caso de Windows, el contenedor de objetos gráficos principal se compone de formas gráficas, mejor conocidas como ventanas, que se encargan de contener elementos gráficos más especializados del entorno, como son:

- Objetos de ejecución de funciones
- Objetos de validación y captura de caracteres
- Objetos de selección de uno entre diversos valores posibles
- Objetos de agrupación y organización de información
- Objetos de captura de datos lógicos
- Objetos de validación y captura de fechas

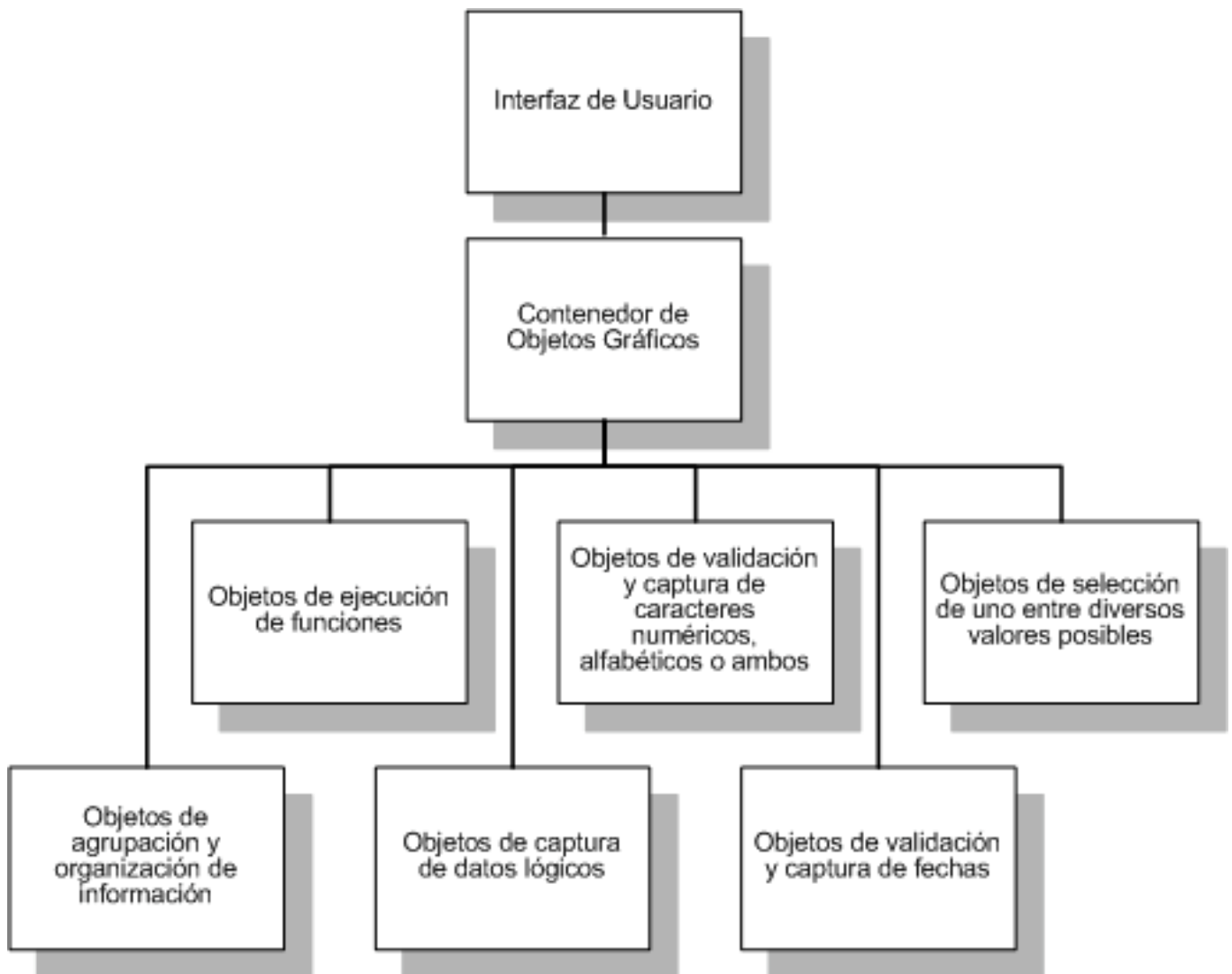


Figura 4.22. Modelo jerárquico de la interfaz.

### *Contenedor de objetos gráficos*

Los contenedores de objetos gráficos son por sí mismos objetos que poseen propiedades y métodos que se encargan de la agrupación de otros elementos. Los contenedores de objetos gráficos como los que se muestran en la figura 4.23 son, en última instancia, hojas en blanco que el programador rellena con controles (objetos), para componer una interfaz de usuario; y con código, para procesar los datos. Para este fin se utiliza un conjunto de objetos completo, escrito con la plataforma .NET Framework, agregando con ello una mayor funcionalidad a estos con el código escrito por el generador de código.

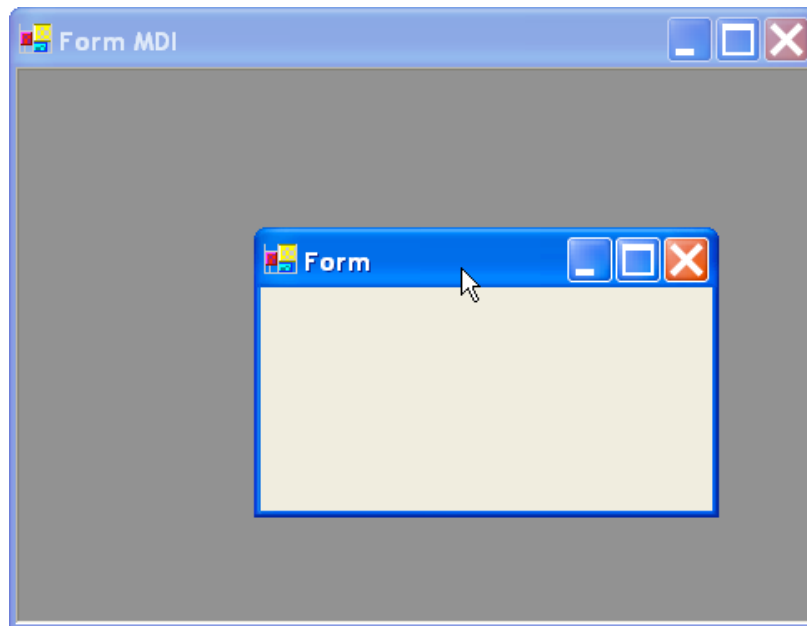


Figura 4.23. Contenedores de objetos gráficos.

#### *Objetos de ejecución de funciones*

Los objetos de ejecución de funciones se encargan de ejecutar acciones asociadas a los mismos, definidas por el sistema en el momento en que el usuario decide su ejecución (Figura 4.24).

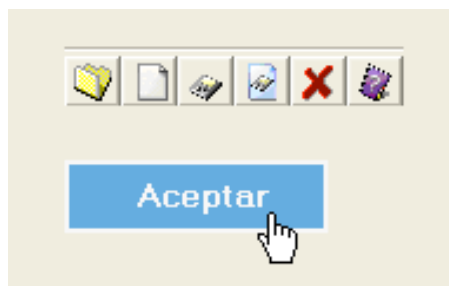
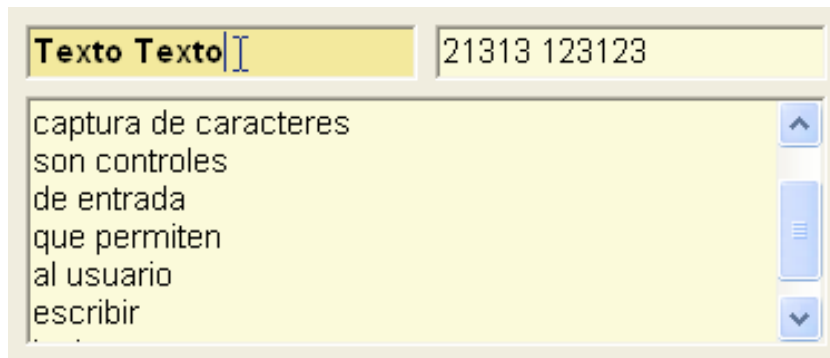


Figura 4.24. Objetos de ejecución de funciones.

#### *Objetos de validación y captura de caracteres*

Los objetos de captura de caracteres son controles de entrada que permiten al usuario escribir texto. Estos objetos se implementan tomando como base los objetos `TextBox` y `RichTextBox` de la plataforma .NET, agregándoles funcionalidad de

validación del tipo, longitud, rango y obligatoriedad del texto contenido en ellos (Figura 4.25).

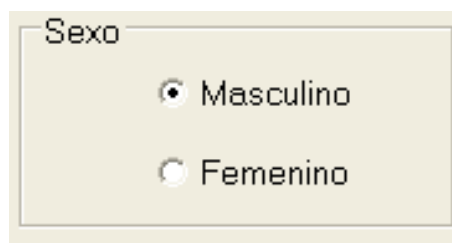


*Figura 4.25. Objetos de captura de caracteres.*

#### *Objetos de selección de uno entre diversos valores posibles*

Estos objetos permiten presentar diversas opciones entre las que se debe seleccionar solamente una de ellas, estos objetos se pueden presentar de dos formas, dependiendo del número de opciones disponibles con que se cuenta:

- **RadioButton.** Estos objetos se presentan normalmente en grupos de botones de opción, entre los que el usuario puede seleccionar uno. Estos objetos se agrupan de forma lógica, por lo que cuando un usuario selecciona un objeto de opción, los demás objetos de la misma clase y del mismo grupo dejan de estar disponibles automáticamente, como se muestra en la figura 4.26. Estos objetos son usados para valores de opciones fijas con un número limitado de valores posibles.



*Figura 4.26. Objetos RadioButton de selección de uno entre varios.*

- **STComboBox.** Estos objetos se basan en la implementación de objetos ComboBox, como se muestra en la figura 4.27, que muestran un campo de

edición combinado que despliega una lista de elementos permitiendo al usuario seleccionar una opción de todas las posibles.

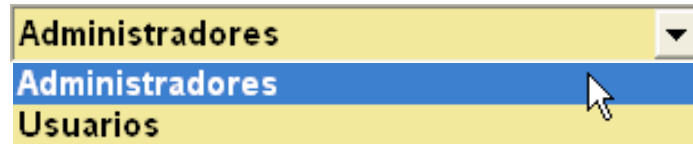


Figura 4.27. Objetos STComboBox de selección de uno entre varios.

### Objetos de agrupación y organización de información

Para la agrupación y la organización del despliegue de los resultados obtenidos sobre una función definida se genera el código de dos clases que heredan propiedades de elementos nativos de la plataforma .NET, que son:

- STContenedor. Este objeto es implementado a través de la herencia de otros objetos, con base en la combinación de objetos Button y Container. Se encarga de ocultar o mostrar las agrupaciones de otros objetos (figura 4.28).

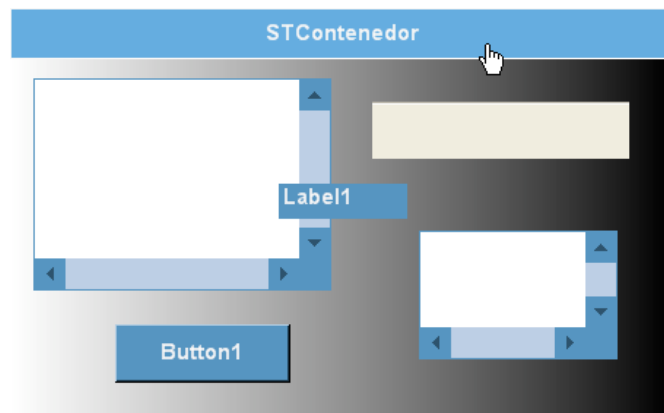


Figura 4.28. Objeto STContenedor.

- STGrid. Este objeto es implementado a través de la herencia de otros objetos, para el despliegue de resultados de forma dinámica que se encarga de mostrar los campos de un origen de datos como columnas en una tabla, en donde cada fila del control representa un registro del origen de datos permitiendo la selección y ordenación de los datos (figura 4.29).

<input type="checkbox"/> Grupo	<input type="checkbox"/> Usuario	Nombre	Apellido Paterno	Apellido Materno
<input type="checkbox"/> Administradores	lvazquez	Luis Antonio	Vázquez	Ramos
<input checked="" type="checkbox"/> Administradores	admin	Luis Antonio	Vázquez	Ramos
<input type="checkbox"/> Administradores	mhernandez	Miguel	Hernandez	
<input type="checkbox"/> Usuarios	evazquez	Eduardo	Vázquez	Castañeda
<input type="checkbox"/> Usuarios	ecalzada	Enrique	Calzada	

Figura 4.29. Objeto STGrid.

### Objetos de captura de datos lógicos

Estos objetos permiten al usuario seleccionar un estado lógico true o false, pueden ser agrupados para presentar varias opciones entre las que el usuario pueda seleccionar una, ninguna o varias.

En la plataforma .NET existen dos objetos en específico para la captura de datos lógicos: los objetos CheckBox, que presentan una marca de verificación cuando se activan; y los objetos CheckedListBox, que son combinaciones de varios objetos CheckBox que aporta funciones prácticas de enlace de datos. Sin embargo, los controles CheckBox individuales permiten un mayor control sobre el diseño (figura 4.30).

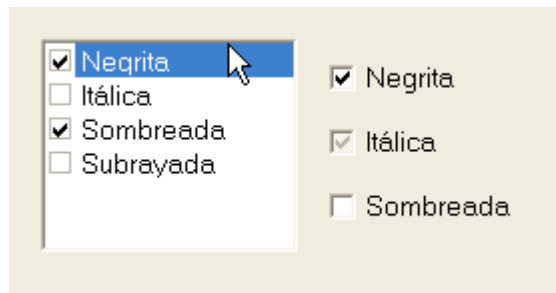


Figura 4.30. Objetos de captura de datos lógicos.

### Objetos de validación y captura de fechas

Los objetos de validación y captura de fechas se basan en la implementación heredada de los objetos DateTimePicker y se utilizan para permitir al usuario seleccionar una fecha y una hora, y para mostrar esa fecha y esa hora en el formato especificado. Se pueden limitar las fechas y las horas que se pueden seleccionar, se

puede crear un estilo de formato propio y la construcción de una cadena de formato personalizado (Figura 4.31).

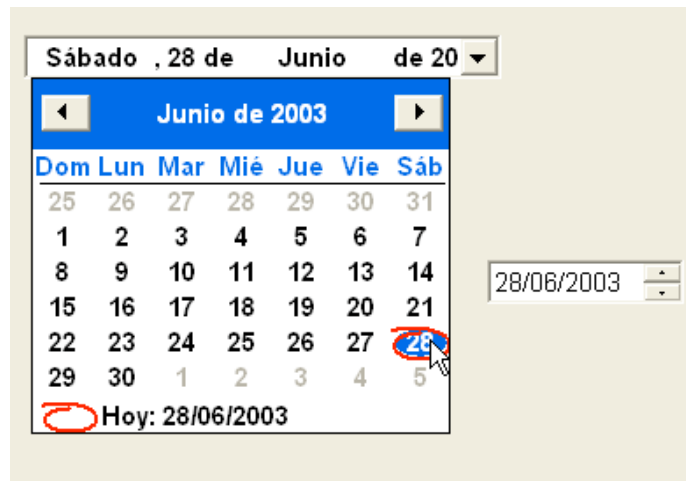


Figura 4.31. Objetos de captura de fechas.

### 4.3.2. Componentes de acceso a datos

Los componentes de acceso a datos se agrupan o dividen en objetos independientes dependiendo del modelo de solución elegido por el usuario, están formados por métodos básicos de consulta, alta, baja y modificación, que se encargan de la manipulación de los datos contenidos en la fuente de información del sistema.

Estos componentes crean cadenas en lenguaje SQL de consulta y modificación, estableciendo cuatro métodos básicos para la manipulación dinámica de la información contenida en la base de datos, que son:

- Consulta
- Alta
- Baja
- Cambio

La descripción del objetivo, el diagrama de flujo y la estructura del pseudo código de cada uno de los componentes de estos métodos, así como los parámetros de entrada y salida que reciben y regresan, es la siguiente:

#### Método de Consulta



El objetivo de este método es realizar la consulta con base en la creación de una instrucción en lenguaje SQL, que se encargue de obtener un conjunto de datos de la base de datos y permitir realizar la selección de una o varias filas o columnas de una o varias entidades.

De acuerdo a los parámetros de entrada, devuelve el resultado de dicha consulta en un objeto *DataTable* que representa un conjunto de datos.

El pseudo código y los diagramas de flujo de este método se presentan en las figuras 4.32 y 4.33 respectivamente, los parámetros de entrada y de salida requeridos por el método *Consulta* son:

- Parámetros de Entrada
  - **conn.** Cadena de Conexión.
  - **tabla.** Nombre de tabla en donde se realizará la consulta.
  - **ligas.** Lista de objetos *Ligas* (tabla, campo, valor, comparación), el cual especifica el nombre de la tabla, el campo, un valor, y el criterio de comparación entre el campo y el valor.
  - **oparams.** Lista de objetos *Parametro* (campo, valor, comparación), el cual especifica el nombre del campo, un valor y la condición de comparación respectivamente.
  - **oparams\_valores.** Lista de objetos *Parametro* (tabla, campo, valor).
  - **olist\_param.** Lista de objetos *ParametroLista* (campo, lista de posibles valores) , el cual especifica el nombre del campo y lista los el rango de valores respectivamente.
  - **v\_criterio.** Valor Booleano que especifica el tipo de unión entre las condiciones que deben reunir los registros a modificar.
- Parámetros de Salida
  - **result.** Devuelve un objeto *DataTable* con el resultado de la consulta.

INICIO DE FUNCION

```

cad_sql = "SELECT ";

FOR EACH ( tabla -> campo IN oCampos)
    cad_sql .= "    tabla.campo , ";
FIN FOR EACH

Elimina_dos_Ultimos_Cars ( cad_sql );
cad_sql .= " FROM tabla"

IF existe lista de objetos ligas entonces
    FOR EACH ( tabla -> liga.lista_params IN ligas)
        ltablasliga += 1;
        FOR EACH ( oparams IN liga.lista_params)
            lparamsliga += 1;
        FIN FOR EACH
    FIN FOR EACH
FIN IF

IF existe lista de objetos oparams entonces
    FOR EACH ( param -> valor IN oparams)
        IF valores diferente de Nulo entonces iparams += 1
    FIN FOR EACH
FIN IF

IF existe lista_params entonces ilista_params = contar(lista_params);
IF ltablasliga > 0 Y lparamsliga > 0 entonces
    FOR EACH ( tabla -> lista_params IN ligas)
        cad_sql .= " INNER JOIN " . tabla . " ON ";
        iparams = cuenta(lista_params);
        FOR EACH ( params -> ( tabla, campo, simb_comp, valor) IN lista_params)
            cad_sql .= tabla.campo . " " . simb_comp . " " . valor ;
            IF iparams > 1 entonces
                IF (v_criterio)
                    cad_sql .= " AND ";
                ELSE
                    cad_sql .= " OR ";
            FIN IF
        FIN IF
    FIN FOR EACH
FIN FOR EACH
FIN IF
FIN IF
FIN IF

```

Figura 4.32. Pseudo código Método Consulta. (Continúa)

```

IF iparams es mayor a 0 ó ilista_params es mayor a 0 entonces
    cad_sql .= " WHERE "
FIN IF

IF iparams es mayor a 0 entonces
    FOR EACH (param->(tabla, campo, simb_comp, valor) In oparams)
        IF existe param->valor entonces
            cad_sql .= tabla.campo . " " . simb_comp . " " . valor ;
        FIN IF
        IF iparams es mayor a 1 entonces
            IF v_criterio es verdadero entonces cad_sql .= " AND ";
            ELSE
                cad_sql .= " OR ";
                iparams -= 1;
            FIN IF
        FIN IF
    FIN FOR EACH
FIN IF

IF ilista_params es mayor a cero entonces
    FOR EACH (olist_param -> (campo, lista_de_valores) IN oParams_listas)
        cad_sql .= campo . " IN (" .
        FOR EACH (elemento IN lista_de_valores)
            sIDs .= elemento . ",";
        FIN FOR EACH
        sIDs = elimina_ultimo_caracter(sIDs);
        cad_sql .= sIDs . ")"
        IF ilista_params es mayor a 1 entonces
            IF v_criterio existe entonces sql_cad .= " AND ";
            ELSE cad_sql .= " OR ";
        FIN IF
        ilista_params -=1;
    FIN FOR EACH
FIN IF

CONEXION_BD( conn );
result = EJECUTA_SQL( cad_sql);
Close( conn );
Regresa( result );

FIN DE FUNCION

```

Figura 4.32. Pseudo código Método Consulta.

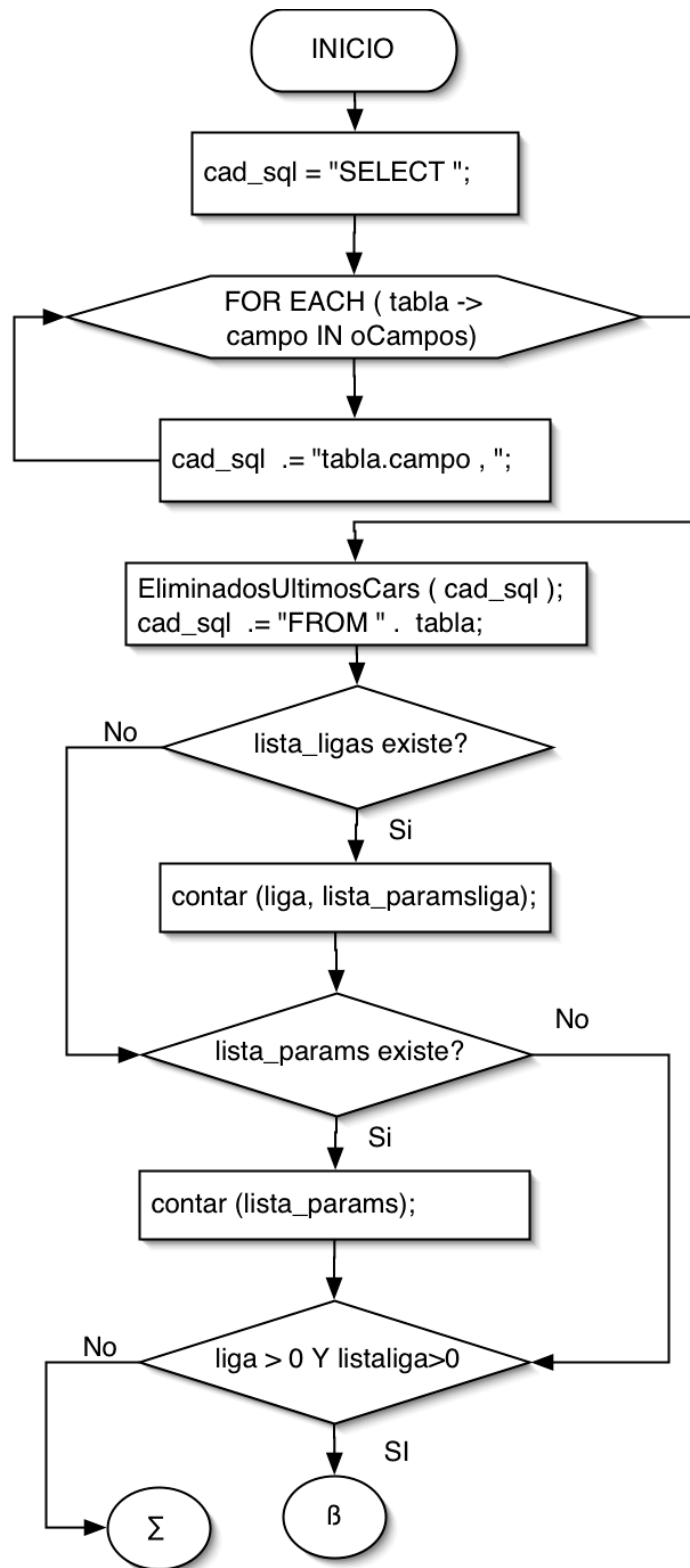


Figura 4.33. Método Consulta. (Continúa)

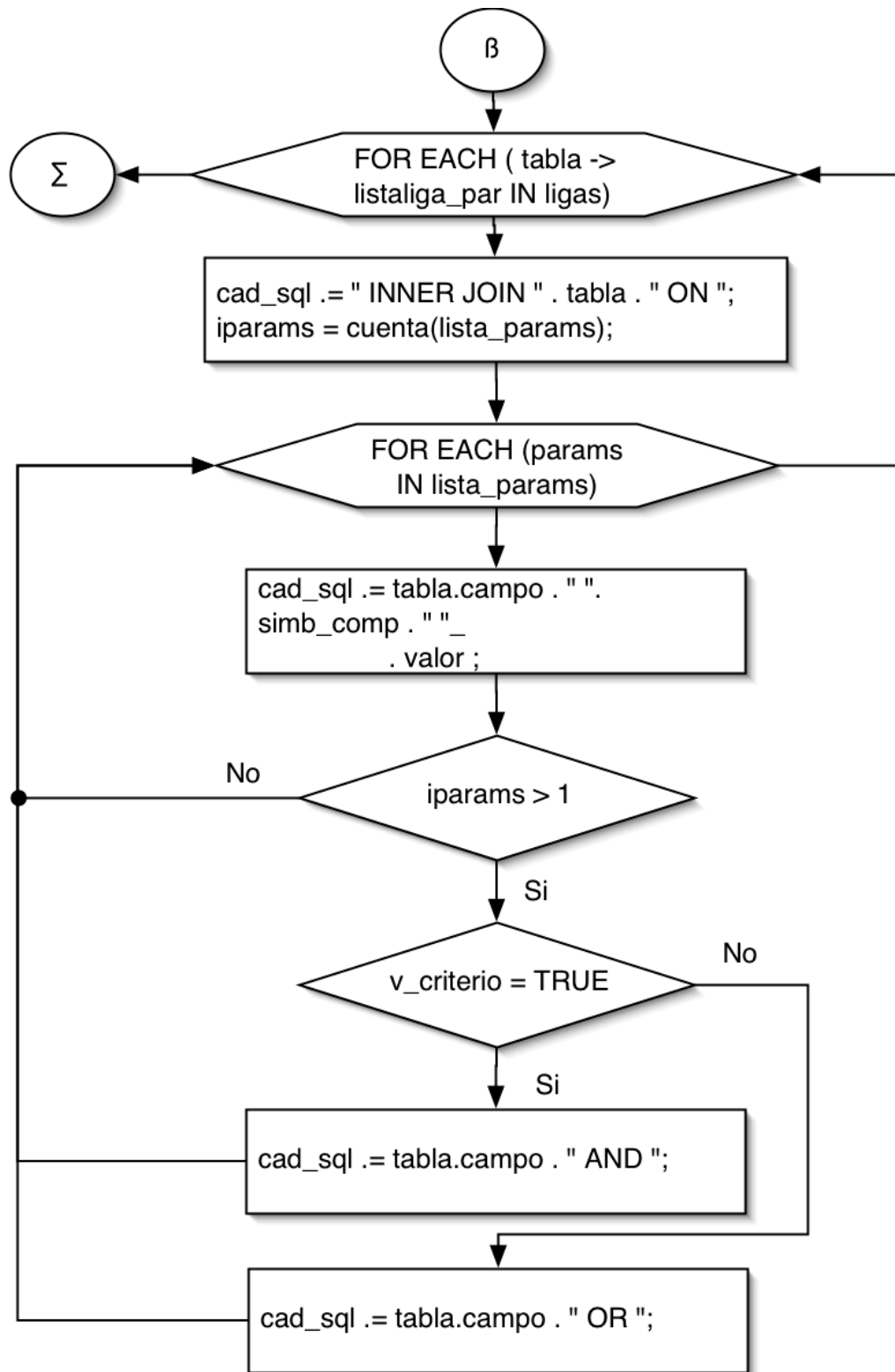


Figura 4.33. Método Consulta. (Continúa)

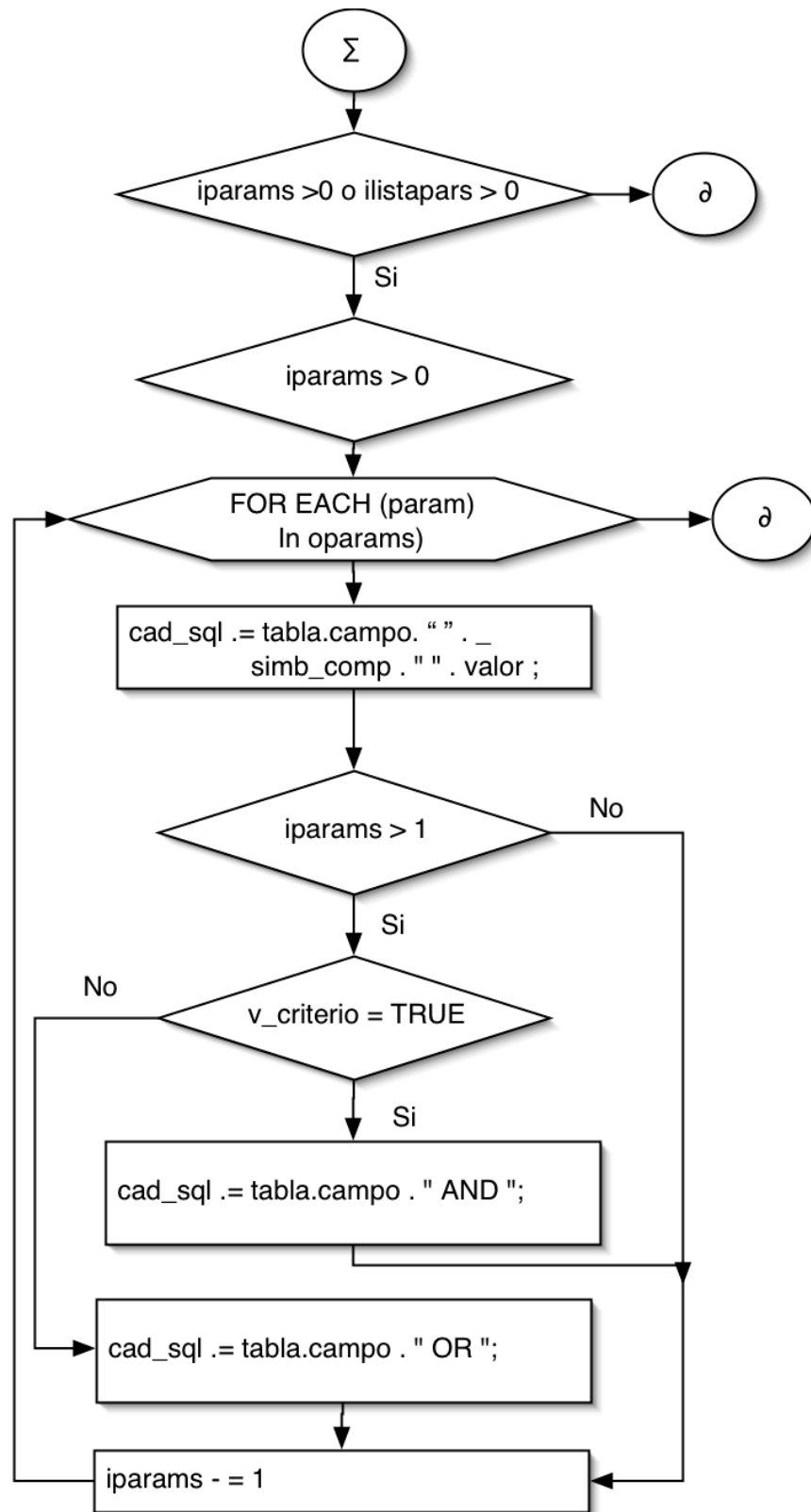


Figura 4.33. Método Consulta. (Continúa)

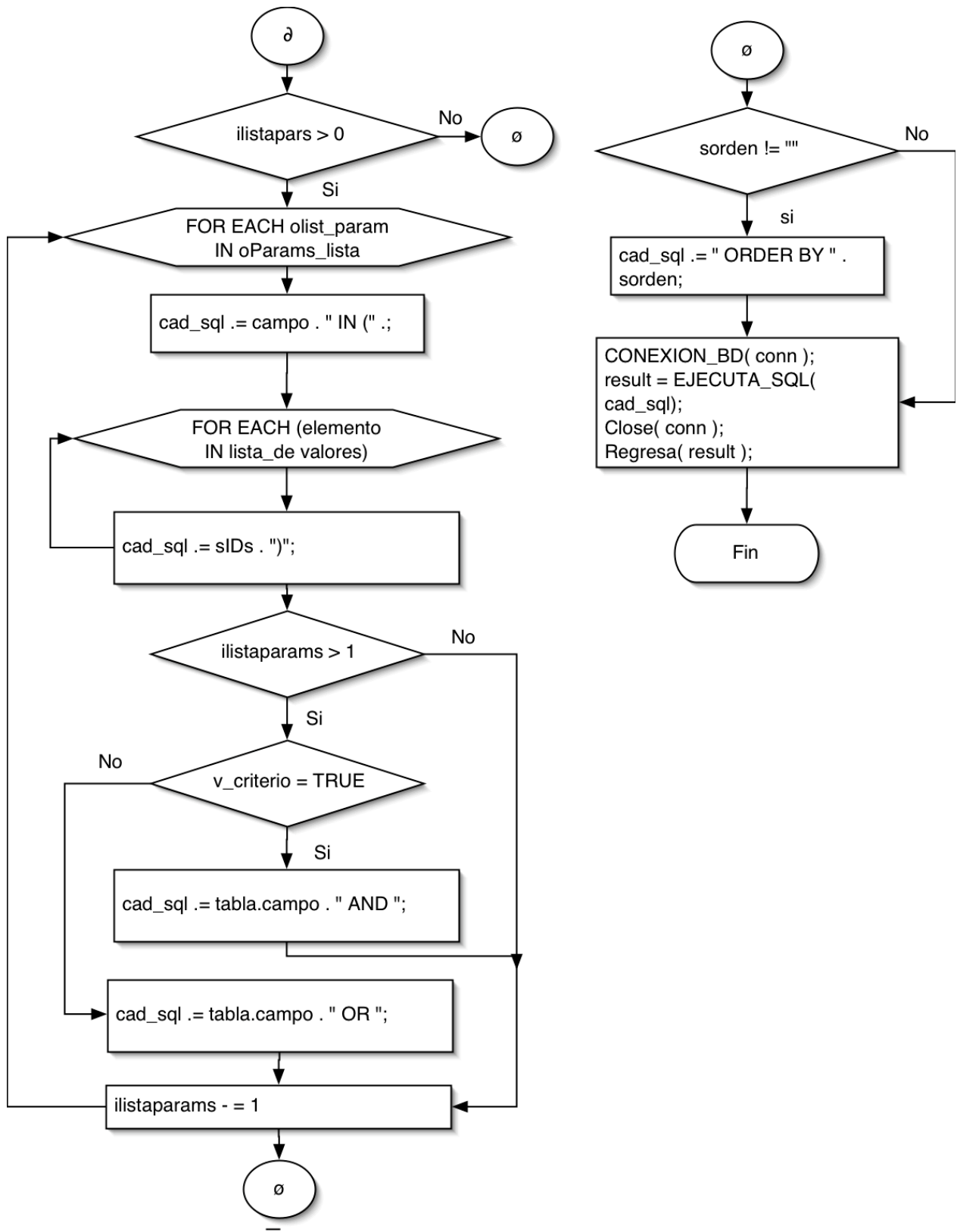


Figura 4.33. Método Consulta.

## Método de Alta

El objetivo de este método es dar de alta un registro en una tabla determinada, el pseudo código de este método se presentan en la figura 4.34 y sus diagramas de flujo en la figura 4.35, los parámetros de entrada y de salida requeridos por el método *Alta* son:

- *Parámetros de Entrada*
  - **conn.** Cadena de Conexión.
  - **tabla.** cadena con el nombre de la tabla.
  - **oparams\_campos.** Lista de objetos *Parametro* (tabla, campo, valor).
  - **oparams\_valores.** Lista de objetos *Parametro* (tabla, campo, valor).
- *Parámetros de Salida*
  - **result.** Devuelve el número de registros afectados.

### INICIO FUNCION

```
// Se consulta la tabla en busca del identificador máximo +1;
id_nuevo =máximo( "id". tabla) + 1;

// Generando la cadena de inserción para un Nuevo registro en leng. Sql.
cad_sql .= "INSERT INTO tabla (id_llave";

FOR EACH param IN oparams_campos
    cad_sql .= " param.campo, ";
FIN FOR

cad_sql = elimina_dos_ultimos_cars(cad_sql);
cad_sql.= " ) VALUES (";

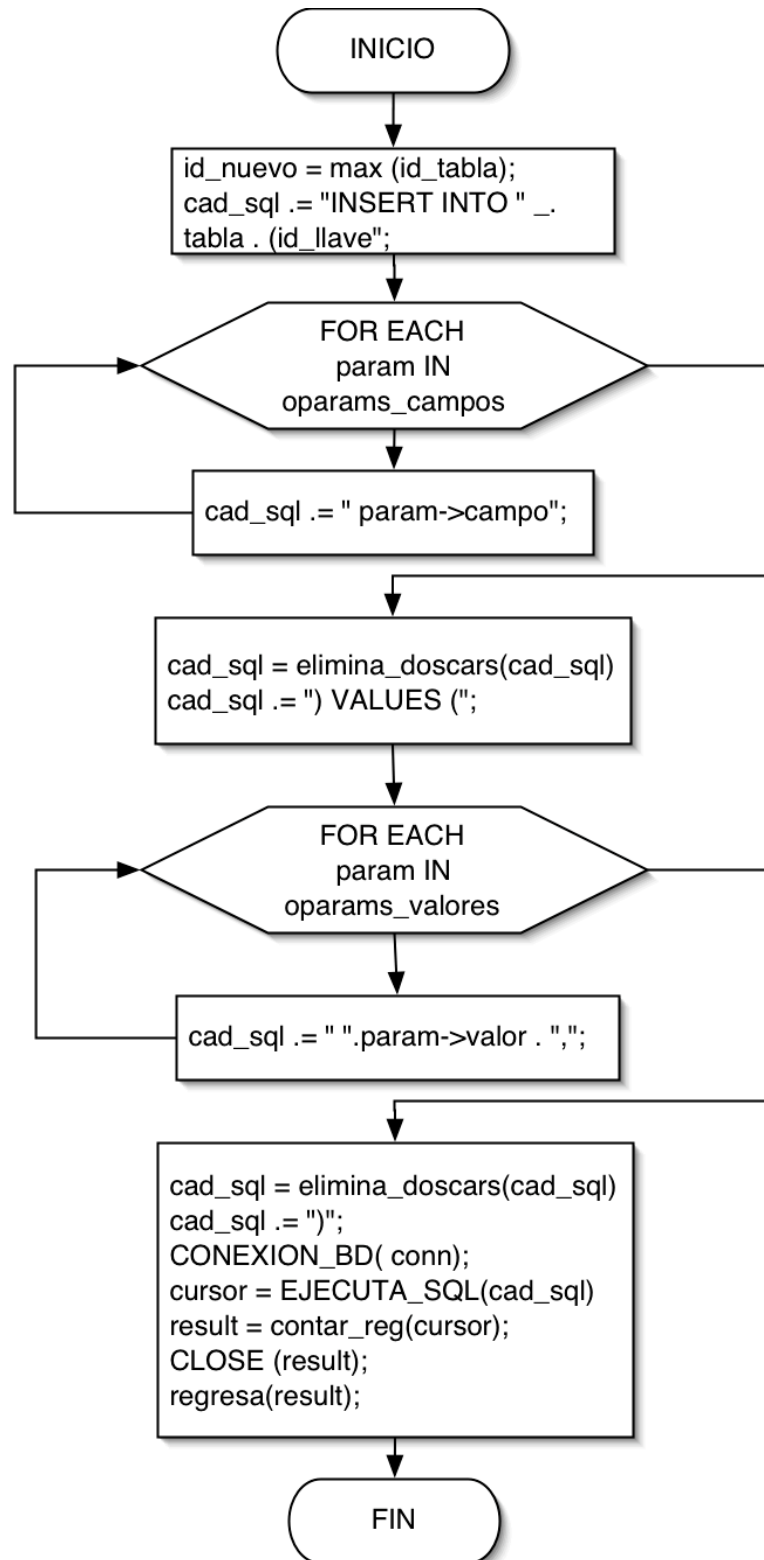
FOR EACH param IN oparams_valores
    cad_sql .= " param->valor, ";
FIN FOR

cad_sql = elimina_dos_ultimos_cars(cad_sql);
cad_sql .= " )";
CONEXION_BD( conn );
cursor = EJECUTA_SQL( cad_sql);
result = cuenta registros(cursor);
Close( conn );
Regresa( result );
```

### FIN DE FUNCION



Figura 4.34. Pseudo código Método Alta.



---

### 4.35. Método Alta.

#### Método Baja

Este método realiza la baja de registros en la base de datos, ya sea de manera lógica (modificando el valor del campo habilitado dentro de las tablas de la base de datos con las que se relaciona el registro) o físicamente (eliminado el registro).

El pseudo código de este método se presenta en la figura 4.36 y sus diagramas de flujo en la figura 4.37, los parámetros de entrada y de salida requeridos por el método *Baja* son:

- *Parámetros de Entrada*
  - **conn.** Cadena de Conexión.
  - **Tabla.** Nombre de tabla en donde se realizará la baja del registro.
  - **v\_criterio.** Operador de unión (AND | OR), valor booleano que especifica el tipo de unión entre las condiciones que deben reunir los registros.
  - **oparams.** Lista de objetos *Parametro* (campo, valor, comparación) el cual especifica el nombre del campo, un valor y la condición de comparación respectivamente.
  - **lista\_oparams.** Lista de objetos *ParametroLista* (campo, lista de posibles valores), el cual especifica el nombre del campo y lista el rango de valores respectivamente.
  - **bllaves.** Valor booleano que indica si se trata de un campo tipo llave principal dentro de la tabla donde se desea eliminar el registro.
- *Parámetros de Salida*
  - **n\_reg\_afectados.** Número de registros afectados

```

INICIO DE FUNCION

CONEXION_DB( conn );
tablas_relacion = BUSCA_RELACION_SQUEMA_BD( tabla );
n_tablas = N_REGISTROS( tablas_relacion );
CLOSE_BD( conn );

IF blaves existe entonces

    cad_sql = "DELETE FROM " . sel->tabla . " WHERE " . sel->campo . IN (" . sIDs . ")";
CONEXION_DB( conn );
cursor = EJECUTA_SQL( cad_sql );
CLOSE_BD( conn );
Reg_afec = N_registros( cursor );

ELSE

    AGREGAR(ocampos, sel );
oconsulta = CON_BD ( conn, tabla, ocampos, , oparametros, oparametrosLista);

    FOR EACH renglon In oconsulta.renglon
        AGREGAR(oIDs, renglon(0) );
    FIN FOR;

    IF cantidad de oIDs es mayor a 0 entonces

        FOR EACH oID IN oIDs

            IF n_tablas es mayor a cero entonces

                FOR EACH renglon IN tablas_relacion.renglon(0)
                    oParams.limpiar()
                    oParams.agregar(sel.tabla, sel.campoid);

                IF num_reg_oconsulta es mayor a cero entonces

                    FOR EACH renglon2 In oConsulta.renglon
                        DesHabilitarIDs[ ] = renglon2;
                    FIN FOR

                ELSE

                    Eliminar[ ] = oID;

                FIN IF

            FIN FOR

        FIN FOR
    
```

Figura 4.36. Pseudo código Método Baja. (Continúa)

```

IF contar (DesHabilitar) es mayor a cero entonces
    FOR EACH element IN DesHabilitar
        SIDs := element . ",";
    FIN FOR

    SIDs = elimina_el_ultimo_cars(SIDs);
    cad_sql = "UPDATE " . tabla . " SET Habilitado = 0 WHERE " . _
    sel->campo . " IN (" . SIDs . ")";

FIN IF

IF contar ( EliminarIDs ) es mayor a cero entonces

    FOR EACH element In EliminarIDs
        sIDs := element . ",";
    FIN FOR

    SIDs = elimina_el_ultimo_cars(SIDs);
    cad_sql = "DELETE FROM " . tabla . "WHERE " . _
    sel->campo . " IN (" . sIDs . ")";

FIN IF

FIN IF

FIN IF

FIN IF

FIN IF

FIN IF

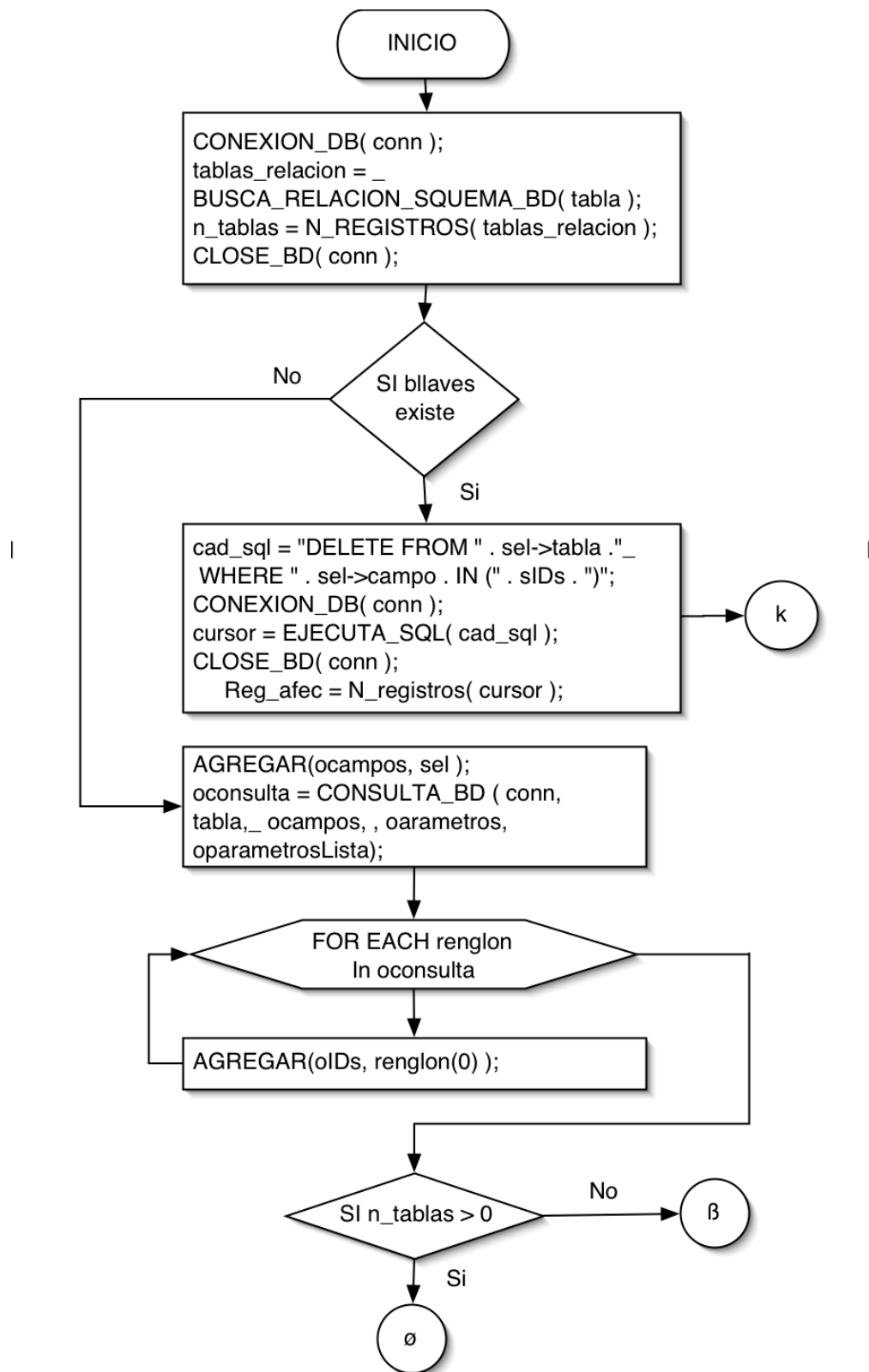
Regresa (n_reg_afectados);

FIN FUNCION

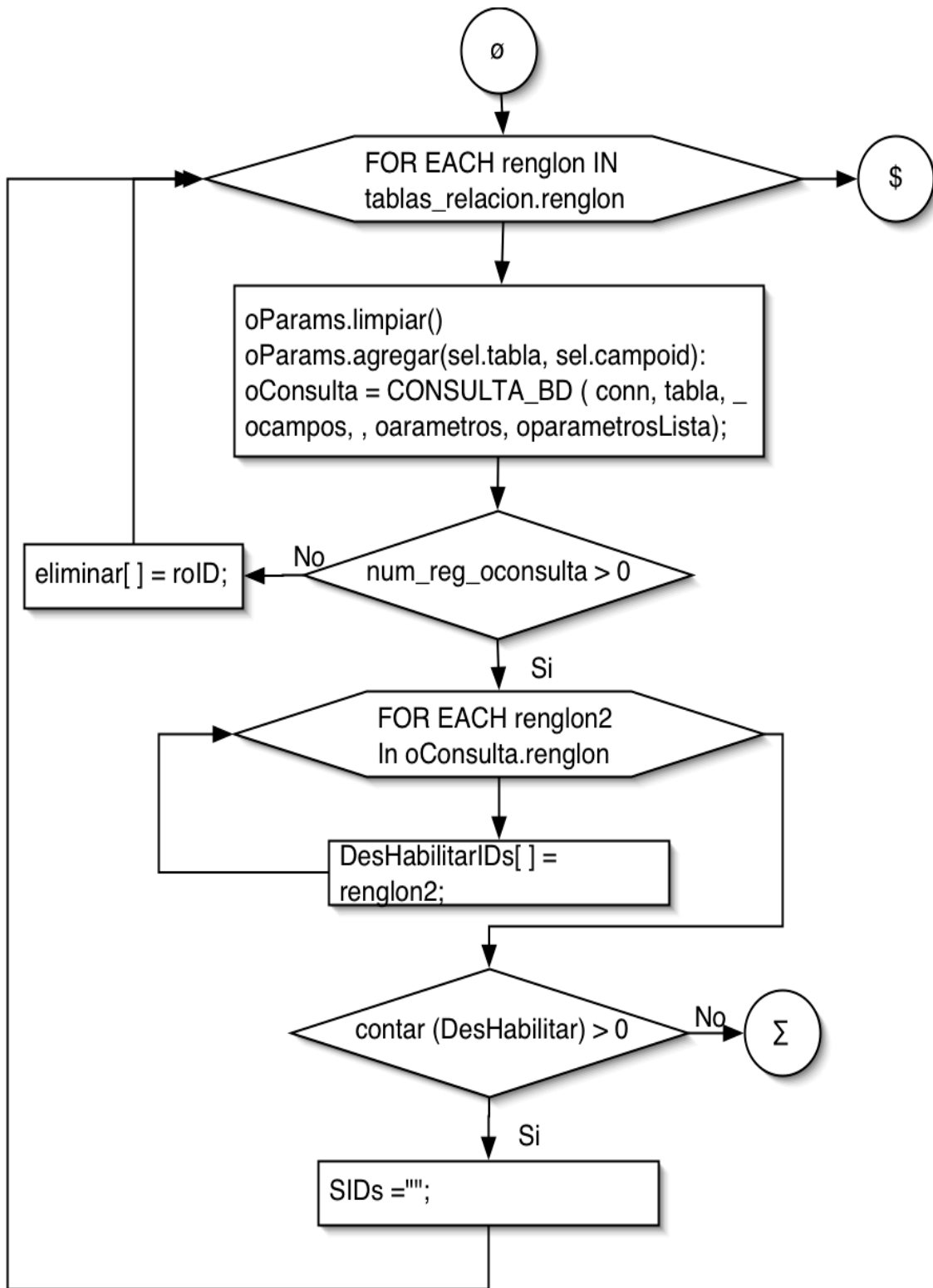
```

Figura 4.36. Pseudo código Método Baja.

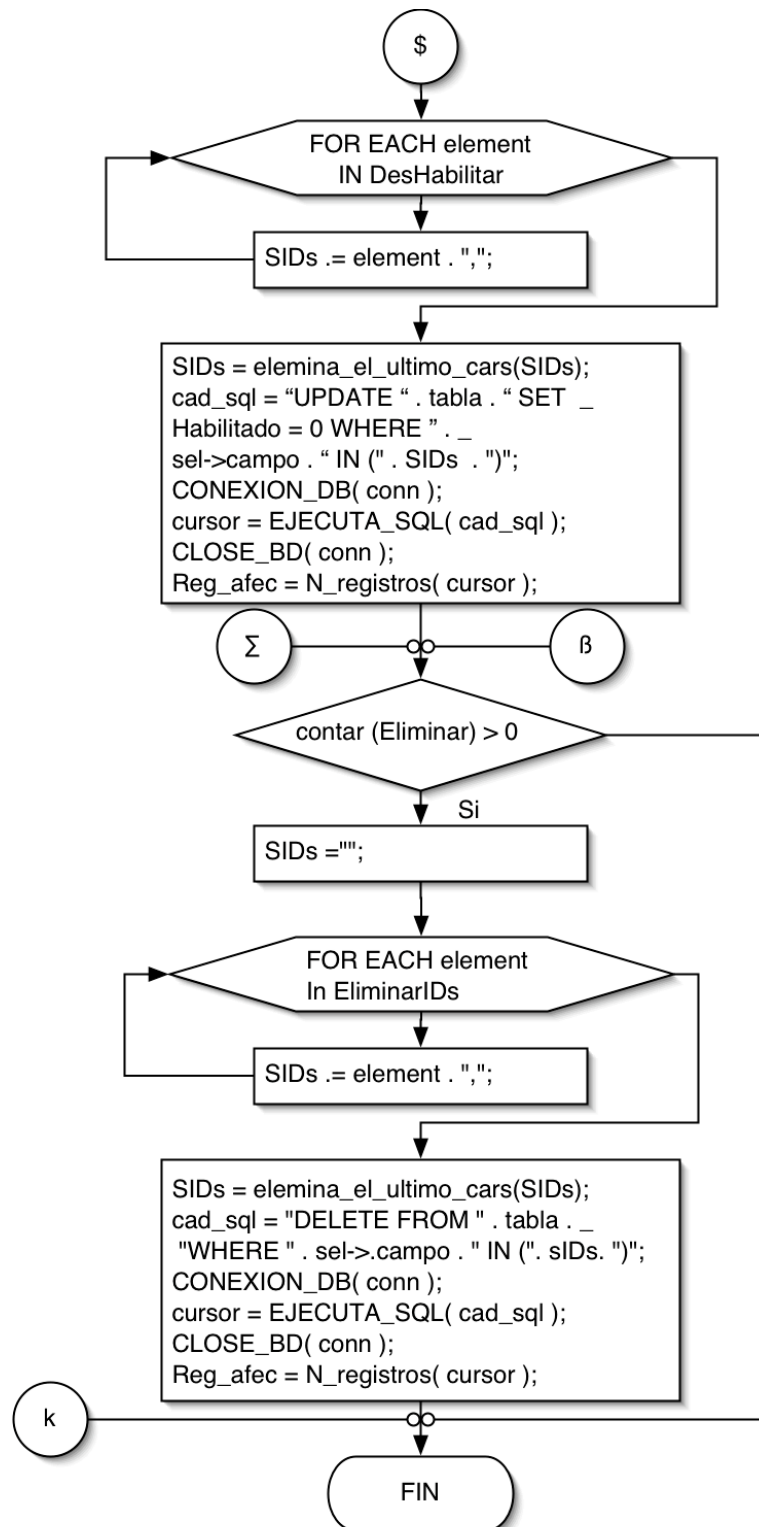




4.37. Método baja. (Continúa)



4.37. Método Baja. (Continúa)



4.37. Método Baja



## Método Modificación

El objetivo de este método es el de generar una instrucción en lenguaje SQL, de acuerdo con los parámetros de entrada, para modificar los valores de los registros, con base en las restricciones especificadas en los mismos parámetros de entrada y ejecutar la instrucción SQL devolviendo el número de registros afectados.

El pseudo código del método se presenta en la figura 4.38 y sus diagramas de flujo en la figura 4.39, los parámetros de entrada y de salida requeridos por el método Modificación son:

### Parámetros de Entrada

- **conn.** Cadena de Conexión.
- **tabla.** Nombre de tabla en donde se modificará(n) el (los) registro(s).
- **v\_params\_valores.** Lista de objetos *Parametro* (campo, valor), el nombre de campo y el valor.
- **v\_criterio.** Valor Booleano que especifica el tipo de unión entre las condiciones que deben reunir los registros a modificar.
- **Oparamas\_condicion.** Lista de objetos *Parametro* (campo, valor), el cual especifica el nombre del campo y el valor que corresponden a las condiciones de búsqueda.

### Parámetros de Salida

- **reg\_afec.** Número de registros afectados.

INICIO DE FUNCION

*Cad\_sql* = "UPDATE " . *tabla* . " SET ";

IF *oparams\_condicion* existe entonces *lparams* = contar (*oparams\_condicion*);

IF *v\_params\_valores* existe entonces

*lparams\_valores* = contar(*v\_params\_valores*);

    FOR EACH *oParams* In *v\_Params\_Valores*

*Cad\_sql* .= *oparams*->campo . " = " . *oparams*->valor . ", ";

    FIN FOR EACH

FIN IF

*Cad\_sql* = eliminadoscars(*cad\_sql*);

IF *iParamsCondicion* es mayor a cero 0 entonces

*Cad\_sql* .= " WHERE ";

    FOR EACH *oParam* In *oParams\_Condicion*

*Cad\_sql* .= *oParam*->Campo. = " . *oParam*->valor

        IF *iParamCondicion* > 1 entonces

            IF *v\_criterio* es verdadero entonces *Cad\_sql* .= " AND ";

        Else

*cad\_sql* .= " OR ";

*iParamCondicion* -= 1

        FIN IF

    FIN IF

FIN FOR

FIN IF

CONEXION\_DB( *conn* );

*cursor* = EJECUTA\_SQL( *cad\_sql* );

CLOSE\_BD( *conn* );

Reg\_afec = N\_registros( *cursor* );

Return (reg\_afec);

FIN FUNCION

Figura 4.38 Pseudo Código Método Modificación.

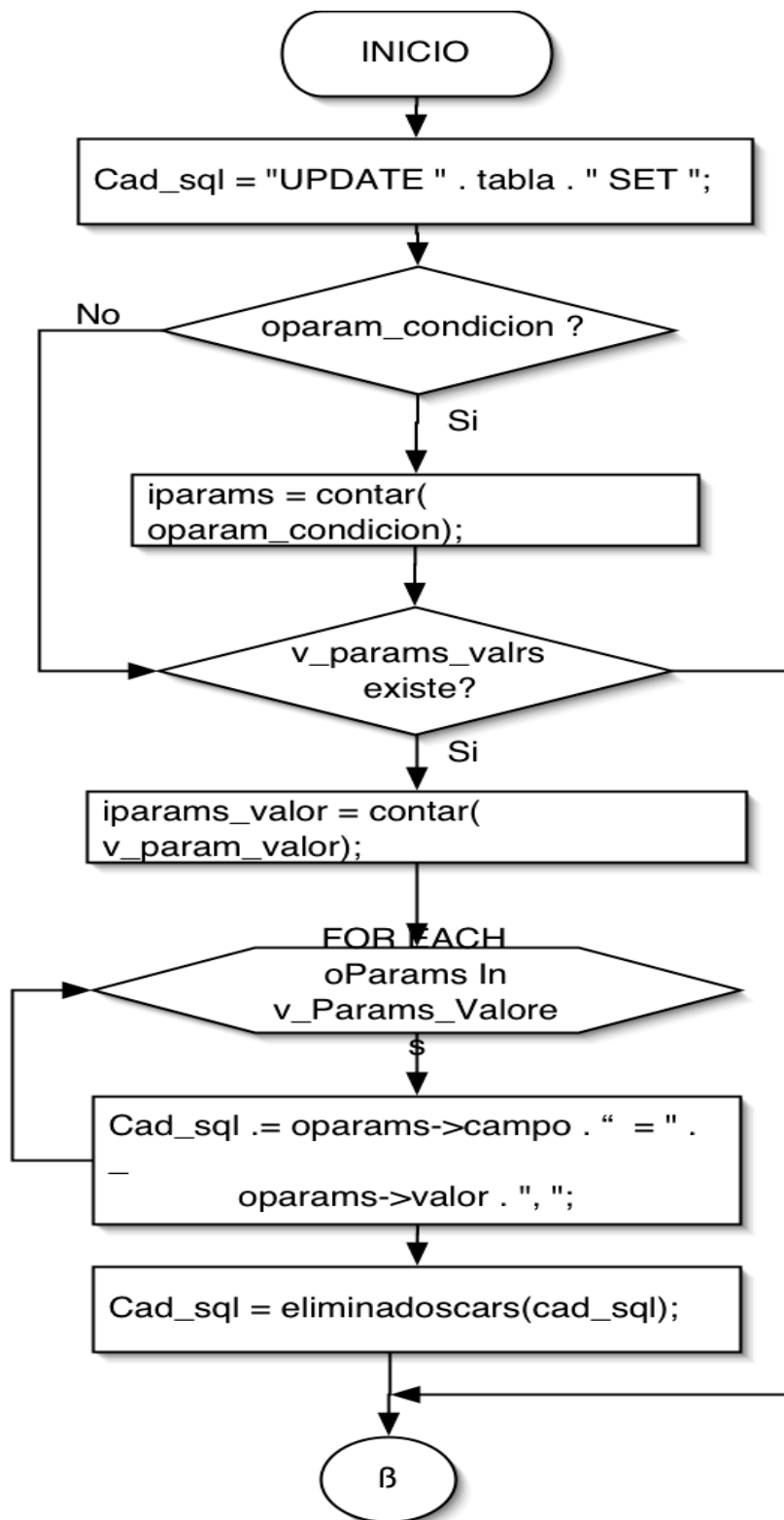


Figura 4.39 Método Modificación. (Continúa)

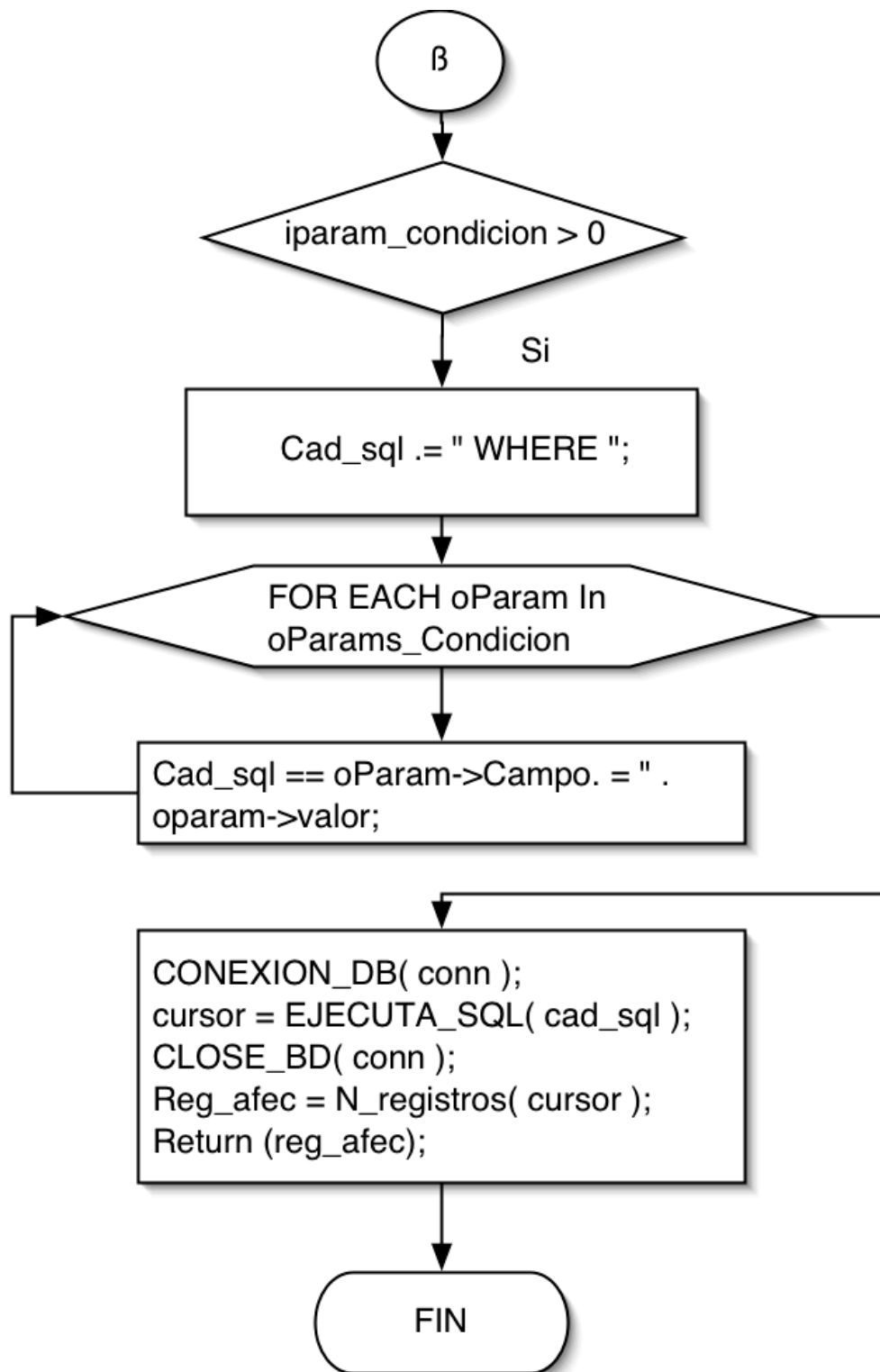


Figura 4.39 Método Modificación.

Una vez definidas las características y funcionalidad del generador de código que utilizaremos para construir el Sistema Administrador de Información, en el capítulo siguiente mostramos la forma en la que la herramienta debe ser configurada para su funcionamiento, así como también la configuración de algunas entidades de acuerdo a las especificaciones determinadas por el usuario y a los requerimientos de hardware y software del sistema.



---

## Desarrollo e Implementación

---

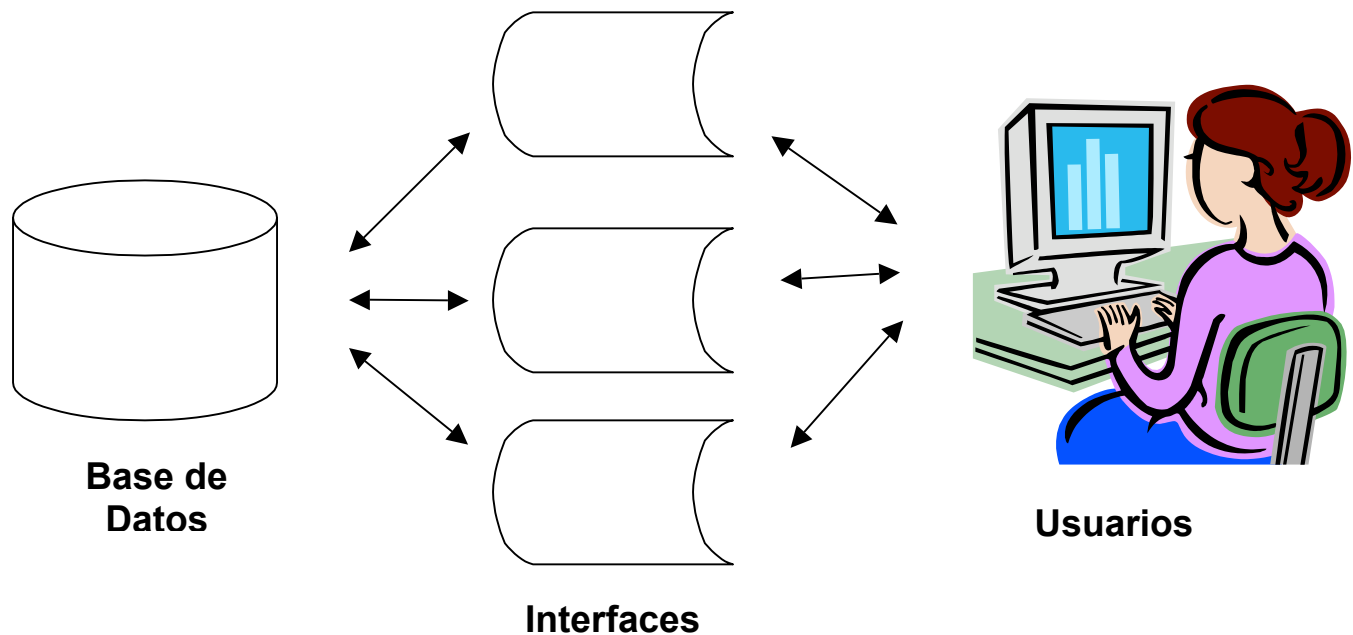
El desarrollo e implementación del sistema se realizará de acuerdo al diseño de los módulos identificados en la base de datos y a la definición de las interfaces recomendadas por el usuario. De esta forma, el desarrollo e implementación consistirá en construir el código fuente de los modelos lógicos que integran cada módulo del sistema, con el objetivo de elaborar un sistema funcional para el usuario de acuerdo a las especificaciones evaluadas.

Las etapas de desarrollo e implementación contemplan principalmente las siguientes tareas:

1. Construcción del sistema de información utilizando el Generador de Código.
2. La implementación de las interfaces del usuario.

El objetivo del sistema es la administración de la información de la Institución Educativa; esta información será manipulada a través de las interfaces, las cuales cuentan con herramientas para la realización de operaciones como la creación, eliminación, consulta y actualización de datos.

El diagrama que se muestra en la figura 5.1. muestra la interacción del usuario a través de las interfaces.



*Figura 5.1. Interacción del Usuario.*

El propósito de las interfaces de usuario es ofrecer un medio de comunicación entre los usuarios y la información almacenada (base de datos), por lo que es muy importante que el sistema sea eficiente, que cuente con interfaces sencillas de utilizar, con un entorno de interacción amigable.

Para la construcción del sistema se implementará la herramienta de generación de código, la cual elaborará el código fuente, a partir de la creación de la base de datos diseñada en el capítulo 3.

## **5.1. Construcción del Sistema de Información**

El generador de código es la herramienta que en nuestro caso generará el código fuente del sistema administrador de información, el cual puede ser implementado una vez que se lleven a cabo las siguientes tareas:

- Creación de la base de datos.
- Configuración del sistema con base a los requerimientos.

La base de datos diseñada previamente es creada, asignándole un espacio físico donde se almacenarán los datos.

Con el propósito de crear el Sistema Administrador de Información se debe llevar a cabo el procedimiento que se detalla en el siguiente diagrama (figura 5.2), utilizando para ello el generador de código.

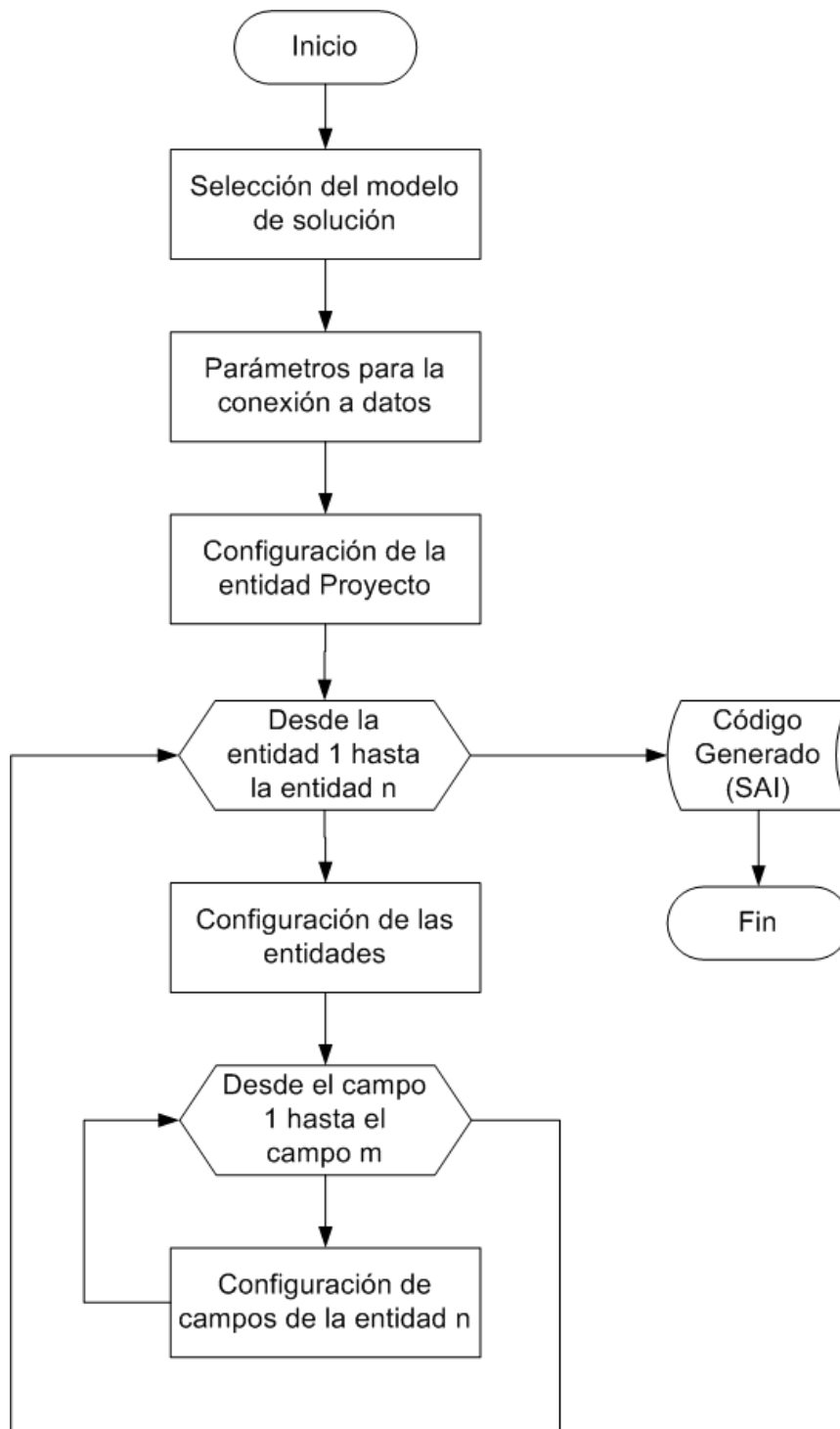


Figura 5.2. Generación del sistema administrador de Información.



Como se muestra en la figura 5.2, es necesaria la selección de un modelo de solución de acuerdo a la infraestructura de software y hardware donde se implementará el sistema y con base en los requerimientos establecidos por el usuario.

Una vez seleccionado el modelo de solución bajo el cual trabajará el sistema, es necesario establecer el origen de datos, así como los parámetros de conexión en los que se basará el generador de código para la construcción del sistema administrador de información.

El paso siguiente es la configuración de los elementos que integran la estructura del sistema administrador de información que son proyecto, entidades y campos de entidades.

La configuración del elemento proyecto tiene como fin establecer las propiedades generales con que contará el sistema administrador de información, que son el modelo de solución, las partes que integran la cadena de conexión a datos y la información sobre el proyecto y el desarrollo del mismo.

La configuración de las entidades es de forma iterativa, estableciendo los valores de las propiedades generales con que cada entidad contará, estas propiedades definen la estructura de los módulos y catálogos que integran el sistema administrador de información, así como el diseño de la interfaz que visualizará el usuario.

Es necesaria la configuración de las propiedades de los elementos campo de cada una de estas entidades, definiendo la forma de validación, el manejo de la estructura relacional, el acceso a los datos, el objeto gráfico y el comportamiento del mismo para la interacción con el usuario.

Al finalizar la configuración del sistema administrador de información se genera el código definido por los elementos que lo conforman.

El código generado permite realizar ajustes, ya sea por requerimientos del usuario o reglas de negocio establecidos por la Institución Educativa antes de la compilación final del sistema.

### **5.1.1. Configuración del Sistema**

Para generar el sistema, es necesario configurar las propiedades de cada uno de los campos de todas las entidades pertenecientes a la estructura de la base de datos dentro del generador de código, sin embargo, mostrar la configuración de cada una de ellas es innecesario debido a la redundancia y excesiva información que esto implica, por ello, decidimos mostrar solo algunos ejemplos que dejan claro la manera en que el generador de código permite la configuración personalizada de cada entidad de acuerdo a los requerimientos especificados.

## Selección del modelo de solución

Al ejecutar el generador de código establecemos la creación de un nuevo proyecto para lo cual se debe seleccionar el modelo de solución en que estará integrado el Sistema Administrador de Información.

Como modelo de solución para el sistema administrador de información se implementará el modelo Cliente–Servidor, ya que así se especificó en los requerimientos que presentó la Institución Educativa; este modelo proporcionará las características necesarias para desarrollar un sistema en el que los usuarios (clientes) puedan acceder desde una PC, y solicitar procesos, los cuales serán ejecutados por el servidor.

## Parámetros para la conexión a datos

La base de datos estará alojada en SQL Server, ya que satisface las necesidades que requiere el sistema, por lo que para la conexión con SQL Server se establecen los parámetros de conectividad con el DBMS, que son: el nombre de la base de datos, el servidor donde se encuentra almacenada esta base y los parámetros de autenticación para el acceso a la información de la misma, si estos son requeridos.

## Configuración de la entidad proyecto

En la figura 5.3. se muestra el módulo general del Proyecto, el cual es la base principal para la configuración del sistema. En el panel de configuración dinámico se especifican las propiedades de la entidad seleccionada, en este caso nos referimos a la entidad raíz proyecto. Algunas de las propiedades de esta entidad se configuran automáticamente al establecer la selección del modelo de solución y la conexión con la fuente de información.

Los valores asociados a las propiedades de la entidad Proyecto son:

- Autor. Esta propiedad es informativa y define el nombre del autor.
- Contraseña. El valor de esta propiedad representa la clave de acceso para la autenticación de la conexión con la base de datos.
- Descripción. Esta propiedad es informativa y especifica la descripción general del proyecto.
- FechaDeCreación. El valor de esta propiedad representa la fecha en la cual fue creado el archivo de proyecto.

- FechaDeUltimaModificación. El valor de esta propiedad representa la fecha de última modificación del archivo de proyecto.
- FuenteDeDatos. El valor de esta propiedad representa el nombre de la base de datos que se utilizará para generar el sistema.
- Modelo. El valor de esta propiedad define el valor del modelo de solución que se implementará en el sistema.
- Nombre. El valor de esta propiedad define el nombre del proyecto.
- Proveedor. El valor de esta propiedad define el nombre del proveedor de datos que se utilizará para la conexión con la base de datos.
- Ruta de Proyecto. El valor de esta propiedad representa la ruta física en la cual se encuentra el archivo de proyecto.
- Servidor. El valor de esta propiedad determina el nombre del servidor de base de datos.
- Usuario. El valor de esta propiedad representa el nombre de usuario para la autenticación de la conexión con la base de datos.

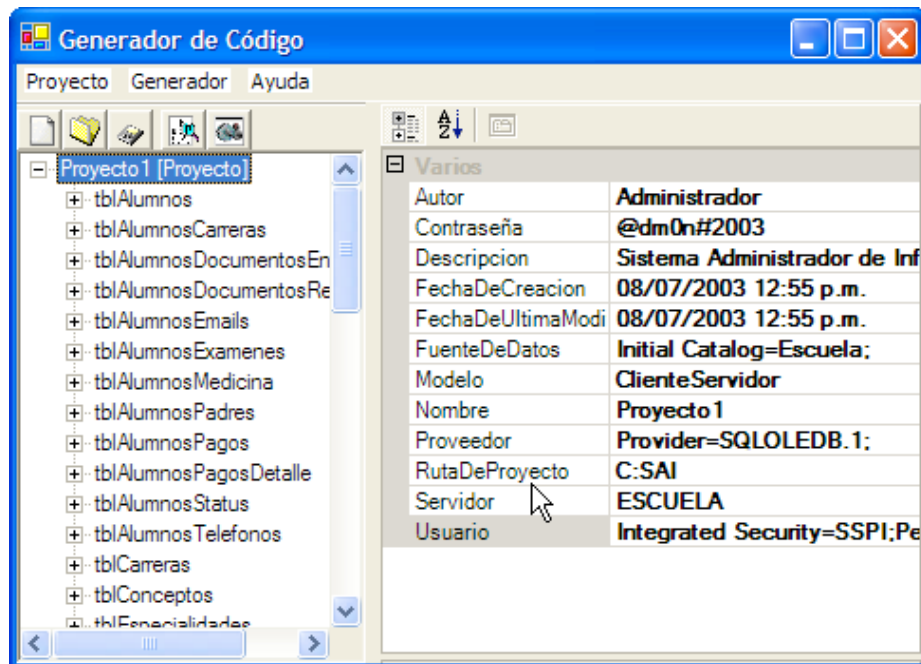


Figura 5.3. Configuración de la entidad proyecto.

En el lado izquierdo de la figura se muestra la estructura de árbol, jerárquica descendente, en donde se observan las entidades pertenecientes al Proyecto, estas entidades, de acuerdo a las especificaciones definidas, deberán ser configuradas sobre sus campos con las propiedades que le corresponden, en caso de ser necesario, ya que algunas propiedades definen su valor por defecto de acuerdo al tipo de dato. El mecanismo de configuración de cada entidad es similar.

Las entidades a configurar para nuestro sistema son: alumnos, padres o tutores, planes de estudio, profesores, pagos, usuarios y los catálogos.

Para esto se debe ir seleccionado una a una las entidades que intervienen en el sistema, al elegirse se mostrará del lado derecho de la ventana el panel con la forma que contiene los campos a configurar. Para entender a detalle este proceso presentamos la configuración de la entidad alumno y algunos de sus campos:

### Configuración de la entidad Alumno

La entidad “alumno” es muy importante dentro de la estructura del sistema administrador de información, ya que contiene la información relacionada a los datos de los alumnos, es por ello que a manera de ejemplo mostraremos la configuración de las propiedades de la entidad que definen al módulo de alumno.

#### *Alumnos*

La figura 5.4. muestra el panel de configuración que nos permite especificar si se desea una pantalla de consulta, modificación y búsqueda. Adicionalmente, el campo EsModulo permite clasificar la funcionalidad de la entidad como módulo o como catálogo del sistema; la propiedad MensajeRegistroExitoso incluye el mensaje que enviará el sistema cuando se efectúe una acción válida sobre la forma de modificación de datos; por último NombreDeTabla indica el nombre físico que tiene la tabla dentro de la fuente de datos que contiene la información de los alumnos.

Varios	
CaptionFormaBuscar	Busqueda de Alumnos
CaptionFormaConsultar	Consulta de Alumnos
CaptionFormaModificar	Nuevo / Modificación Alumnos
EsModulo	True
MensajeRegistroExitoso	Función efectuada exitosamente
NombreDeTabla	tblAlumnos

*Figura 5.4. Configuración de la entidad alumnos.*

Una vez terminada la configuración de las propiedades de la entidad alumnos, se deben configurar los campos de la misma, los cuales incorporan nuevos elementos asociados a todas sus propiedades, que pueden ser utilizados y personalizados de forma flexible pero con valores específicos, estos elementos son los siguientes:

- **CaptionControl.** Especifica la etiqueta que se presentará en la ventana como descripción.
- **EsCampoHabilitado.** Indica si se trata del campo que especifica, por su valor booleano, el borrado lógico de datos.
- **EsFotografia.** Este campo permite indicar si el valor de la propiedad es una ruta que apunta a un archivo de imagen, limitando así el dominio de valores que puede contener la propiedad.
- **EsRequerido.** Indica si este campo será obligatorio para la captura.
- **Longitud.** Indica el número de caracteres permitidos para este campo.
- **MensajeDeValidacion.** Especifica el mensaje que se desplegará al usuario en caso de un error en el tipo de dato capturado.
- **MostraEnBusqueda.** Si está en *True*, permite que este campo se visualice en la interfaz de búsqueda.
- **MostraEnConsulta.** Si está en *True*, permite que este campo se visualice en la interfaz de consulta.
- **MostraEnModificacion.** Si está en *True*, permite que este campo se visualice en la interfaz de modificación.
- **NombreDeCampo.** Especifica el nombre del campo en la tabla de la base de datos.
- **OrdenarEnConsulta.** Si está en *True*, permite ordenar la información por los valores de este campo en la interfaz de consulta.
- **PosicionEnLaForma.** Especifica el orden en que se visualizará el control asociado a este campo en la forma.
- **ValorMaximo.** Indica el valor máximo que tendrá este campo (sólo se utiliza si el valor es de tipo numérico) .
- **ValorMinimo.** Indica el valor mínimo que tendrá este campo (sólo se utiliza si el valor es de tipo numérico).
- **Combo Box.** Especifica una sentencia SQL, en donde esa sentencia al ser ejecutada formará una lista con los valores posibles.
- **OptionButtons.** Especifica con base en duplas (etiqueta, valor) las opciones del campo.

Como parte del proceso de configuración “alumnos”, se deben configurar las propiedades que le corresponden, en este apartado se explican algunas de las propiedades con relación al modo de configuración para ejemplificar el proceso:

### Configuración del campo IDalumno

Como este campo es un identificador propio de la entidad, no es modificable y no es necesaria la consulta y búsqueda del mismo, por lo que a los valores de las propiedades *MostraEnBusqueda*, *MostraEnConsulta* y *MostraEnModificación* se asignan el valor *False*, las demás propiedades permanecerán con los valores por defecto. El resultado de la configuración de este campo se muestra en la siguiente figura:

Varios	
CaptionControl	Identificador de Alumno
EsCampoHabilitado	False
EsFotografia	False
EsRequerido	True
Longitud	10
MensajeDeValidacion	
MostraEnBusqueda	True
MostraEnConsulta	True
MostraEnModificacion	False
NombreDeCampo	IDAlumno
OrdenarEnConsulta	False
PosicionEnLaForma	0
ValorMaximo	0
ValorMinimo	0

Figura 5.5. Configuración del campo IDalumno.

### Configuración del campo Número de cuenta

La configuración del campo Número de cuenta se muestra en la figura 5.6, en la cual se especifica que el campo será mostrado en las interfaces de consulta, búsqueda y modificación.

Varios	
CaptionControl	Número de cuenta
EsCampoHabilitado	False
EsFotografia	False
EsRequerido	True
Longitud	15
MensajeDeValidacion	
MostraEnBusqueda	True
MostraEnConsulta	True
MostraEnModificacion	True
NombreDeCampo	Número de cuenta
OrdenarEnConsulta	False
PosicionEnLaForma	0
ValorMaximo	0
ValorMinimo	0

Figura 5.6. Configuración del campo Número de cuenta.

### Configuración del campo sexo

La configuración del campo sexo, el cual está definido por un número limitado de valores, se muestra en la interfaz de alta / modificación a través de controles de tipo option buttons, como se muestra en la figura 5.7.

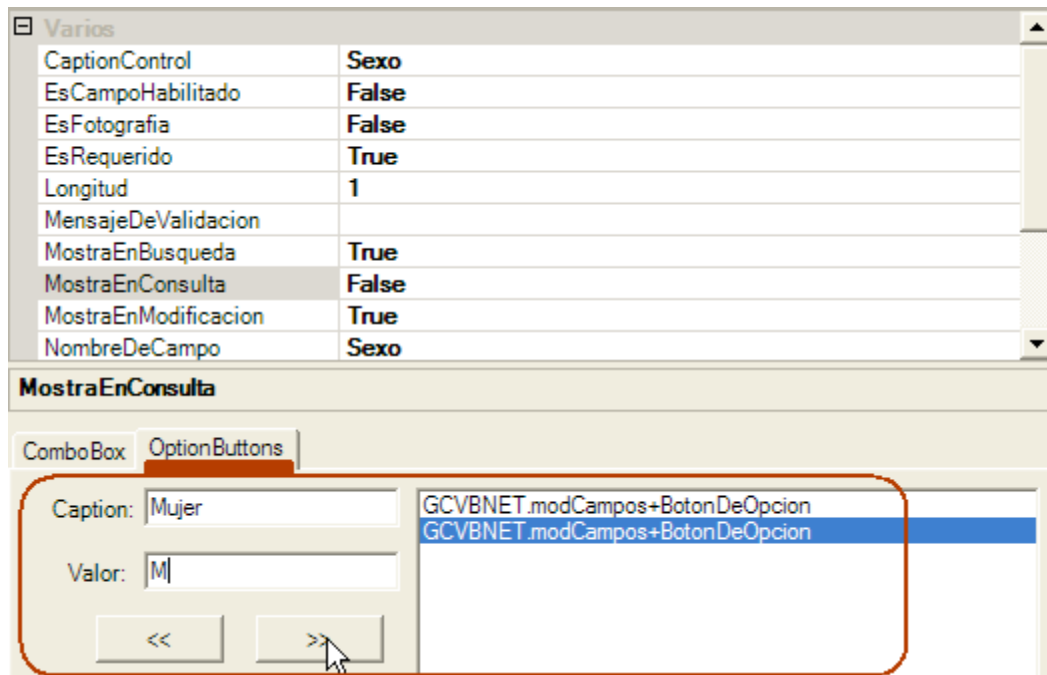
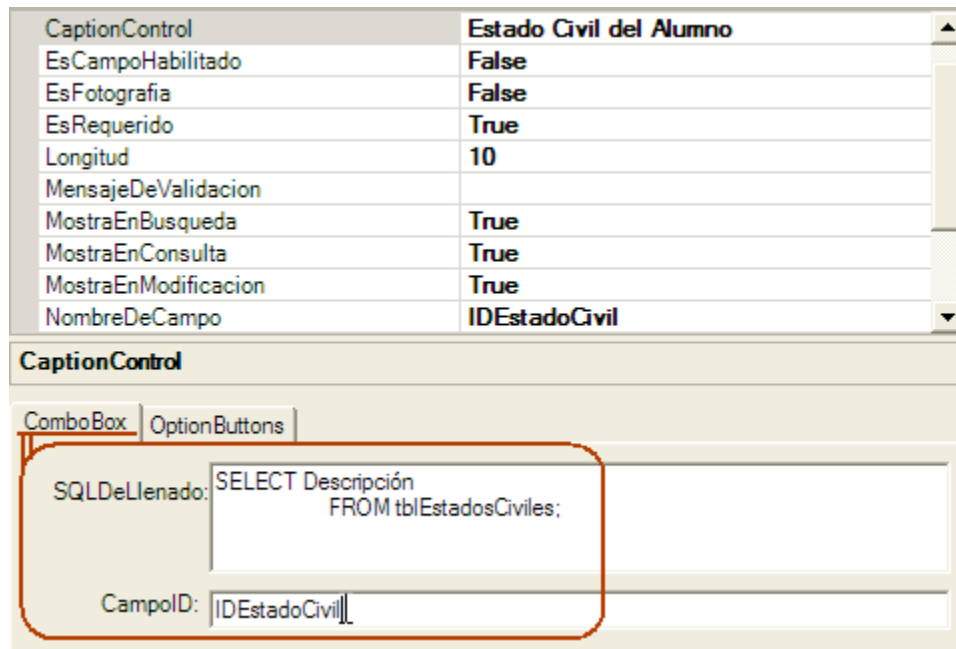


Figura 5.7. Configuración del campo sexo.

### Configuración del campo IDEstadoCivil

En la figura 5.8. se muestra la configuración del campo IDEstadoCivil, para establecer el rango de valores se hace uso de la propiedad ComboBox, que como se puede observar en la figura contiene una consulta SQL, la cual permitirá visualizar un listado de los tipos de estado civil que estén dados de alta en el catálogo para las interfaces de alta / modificación.



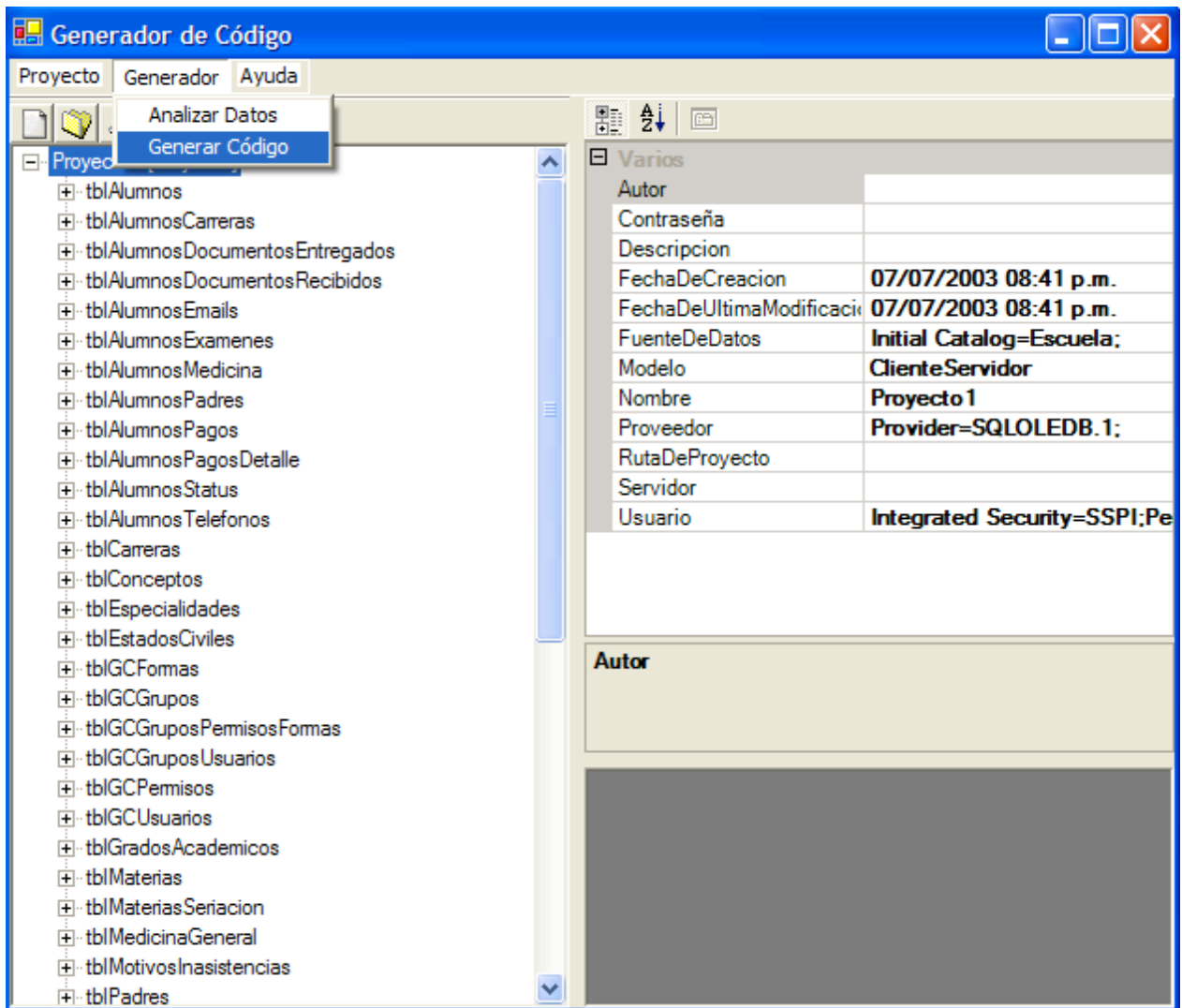
*Figura 5.8. Configuración del campo IDEstadoCivil.*

Las entidades restantes se configuraron de forma similar a las mostradas anteriormente. Una vez que se termine la configuración de todas las entidades, se utiliza la herramienta para generar el Sistema Administrador de Información, como se muestra en la figura 5.9.

Al presionar sobre la opción “Generar Código” del menú del “Generador de Código” se genera el código fuente, el cual contiene las bases del sistema, de las cuales podemos mencionar:

- Creación de clases. Creación de objetos y clases comunes a la aplicación, tales como objetos de usuario, ventanas, etc.
- Creación de ventanas. Contenedor de objetos de interfaz y controles.
- Creación de controles. Dispositivos que realizan las funciones que el usuario desee utilizar.
- Codificación. Por medio de programación indica el comportamiento de objetos y controles.





*Figura 5.9. Generación de Código.*

Una vez generado el código, se pueden agregar nuevos módulos al proyecto que cumplan con reglas de negocio específicas, ya que estos módulos no pueden ser creados por el Generador de Código.

Después de realizar los pasos que se han descrito anteriormente, se concluye con la generación del sistema. En adelante se describirá el funcionamiento de las interfaces que integran al sistema.

## 5.2. Descripción gráfica del sistema

El sistema está formado por una serie de interfaces que proporcionan la funcionalidad y cubren los requerimientos y especificaciones antes descritas, estas interfaces son utilizadas para interactuar con el usuario, realizando por medio de

eventos, operaciones de carácter administrativo. La figura 5.10. muestra la estructura de interfaces al arrancar el sistema, y el menú de la interfaz principal:

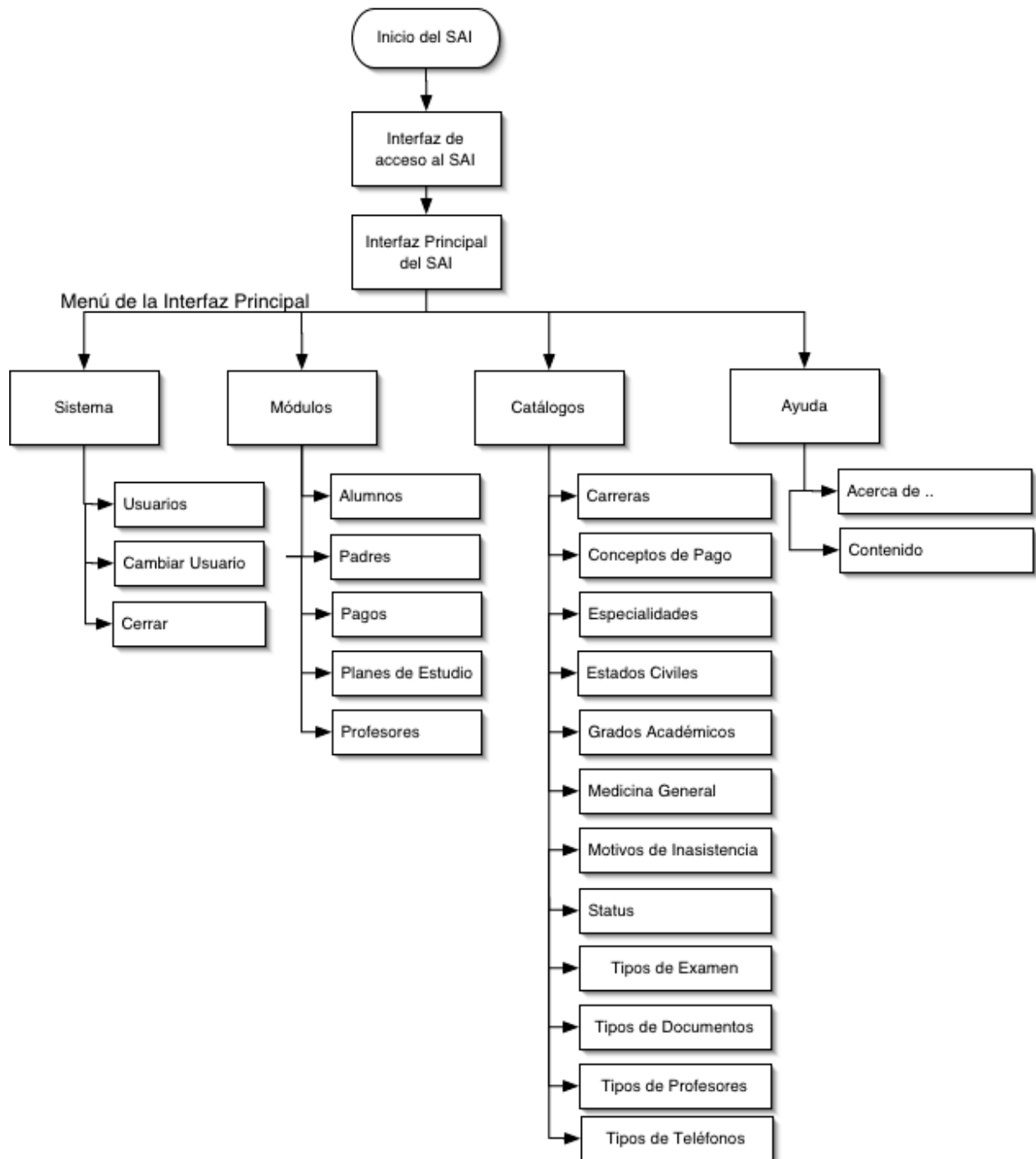


Figura 5.10. Inicio del Sistema.

En la figura anterior se observa que el sistema cuenta con una interfaz inicial, la interfaz de acceso a las funciones principales y la estructura del menú de la interfaz principal.

### **Interfaz de inicio**

Al iniciar el sistema, se presenta la pantalla que se muestra en la figura 5.11, la cual tiene el símbolo que representa a la institución y el nombre del sistema.

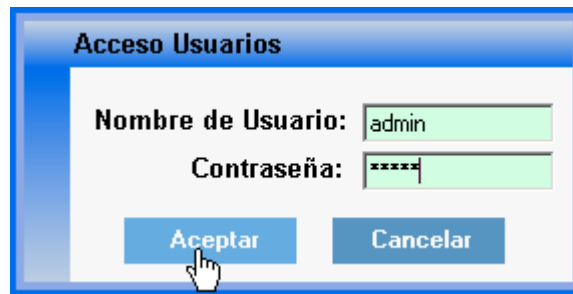


*Figura 5.11. Pantalla de inicio.*

Esta pantalla se presentará mientras que los controles y recursos del sistema están siendo cargados en memoria, esta pantalla se cerrará automáticamente dando paso a la interfaz de acceso.

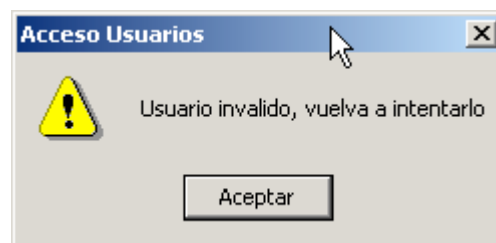
### **Interfaz de Acceso**

La interfaz de acceso se muestra en la figura 5.12, esta interfaz solicita el nombre de usuario y contraseña para acceder al sistema.



*Figura 5.12. Pantalla acceso al sistema.*

En caso de que los datos introducidos en los campos sean incorrectos, se mostrará un mensaje de advertencia que indica que los datos son inválidos, como se muestra en la figura 5.13.



*Figura 5.13. Mensaje de Error en la interfaz de acceso*

El sistema solamente dará la oportunidad de equivocarse en dos ocasiones, en la tercera ocasión la aplicación se cerrará por completo como medida preventiva.

En caso de que los datos dentro de los campos sean correctos desplegará la ventana principal.

### **Interfaz Principal**

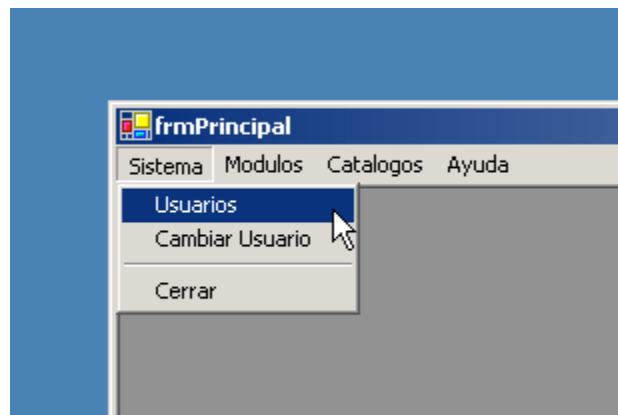
La interfaz principal presenta un menú con las opciones a las cuales se tiene permiso de entrar, de acuerdo al perfil definido para el usuario. En su totalidad las opciones de menú son las siguientes:

- Sistema
- Módulos
- Catálogos
- Ayuda

Estas opciones a su vez cuentan con submenús, los cuales se detallan a continuación:

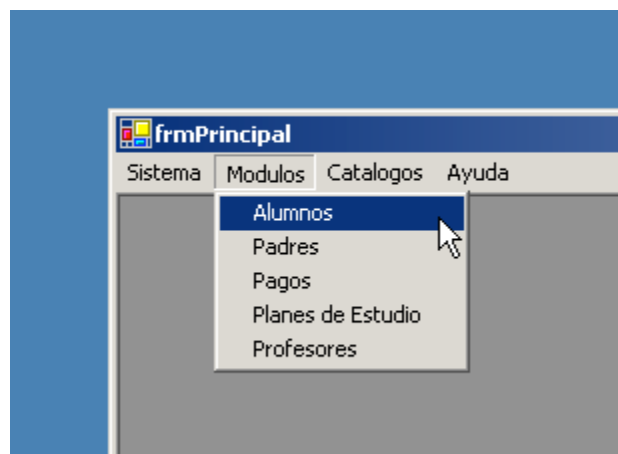
La figura 5.14 muestra el menú sistema, el cual presenta las opciones Usuarios, Cambiar Usuario y Cerrar, que se describen a continuación:

- Usuarios. Esta opción permite el acceso al modulo de usuarios, en el cual se administra las altas, bajas y modificaciones de grupos, usuarios y permisos del sistema.
- Cambiar Usuario. Esta opción cierra la sesión del usuario activo, permitiendo ingresar al sistema con otro usuario.
- Cerrar. Esta opción permite salir del sistema.



*Figura 5.14. Menú Sistema.*

La figura 5.15 muestra el menú módulos, el cual presenta las opciones Alumnos, Padres, Pagos, Planes de Estudio y Profesores.

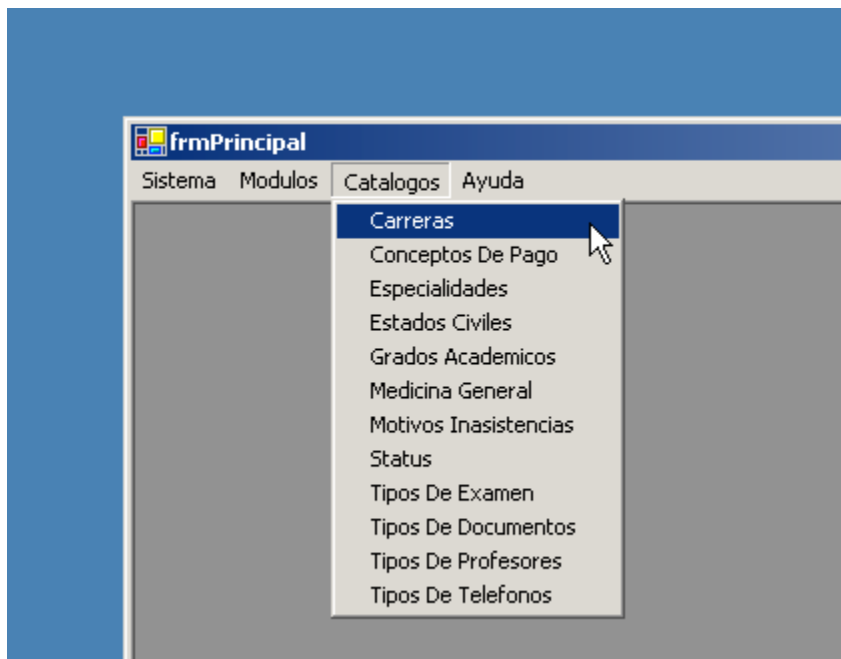


*Figura 5.15. Menú Módulos.*

Las opciones del menú módulos se describen a continuación:

- Alumnos. La opción Alumnos permite el acceso al módulo en el cual se administra la información relacionada con los alumnos.
- Padres. Esta opción permite el acceso al módulo en el cual se administra la información relacionada a los padres o tutores de los alumnos.
- Pagos. La opción pagos permite el acceso al módulo de pagos, en el cual se administra la información referente a pagos.
- Planes de estudio. Esta opción permite el acceso al módulo de administración de planes de estudio.
- Profesores. La opción profesores permite el acceso al módulo profesores, en el cual se administra la información relacionada al mismo.

La figura 5.16 muestra el menú Catálogos, el cual contiene todas las opciones de los catálogos del sistema.



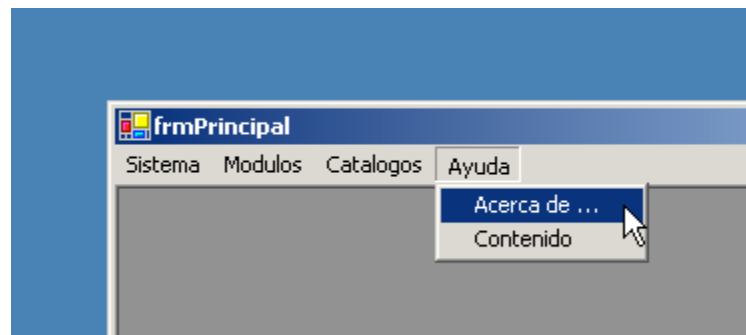
*Figura 5.16. Menú Catálogos.*

El menú de catálogos al ser seleccionado muestra el listado de catálogos pertenecientes al sistema; la definición de los catálogos se especificaron en la etapa de diseño dando como resultado los siguientes elementos:

- Carreras. Al elegir esta opción se muestra la interfaz de consulta donde se listan las distintas carreras que se imparten dentro de la institución.

- Conceptos de Pago. Seleccionando esta opción se podrá modificar el contenido de dicho catálogo.
- Especialidades. Lista los distintos tipos de especialidad en los que está incluido el profesor, lo cual permitirá modificar el contenido del catálogo.
- Estados Civiles. Lista los distintos estados civiles que puede adoptar un alumno o profesor en un momento determinado.
- Grados Académicos. Lista los grados académicos que puede alcanzar el alumno dentro de la institución, en caso de existir un nuevo grado se podrá ampliar.
- Medicina General. Esta opción permite administrar el catálogo de Medicina General.
- Motivos de Inasistencias. Permite modificar el catálogo de motivos de inasistencia.
- Status. Catálogo que presenta las distintas situaciones especiales en las que se puede incurrir.
- Tipos de Examen. Permite la modificación de la lista de los tipos de examen.
- Tipos de Documentos. Permite la modificación de la lista de los tipos documentos que deben ser presentados por alumnos o profesores.
- Tipos de profesores. Permite la modificación de la lista de tipos de profesores que existen dentro de la institución.
- Tipos de Teléfonos. Permite la modificación de la lista los distintos medios por los cuales se puede contactar a un alumno o profesor.

La figura 5.17 muestra el menú Ayuda, el cual contiene las opciones de ayuda que ofrece el sistema.



*Figura 5.17. Menú Ayuda.*

El menú Ayuda despliega dos opciones:

- Acerca de. Muestra en una ventana la información general del sistema.
- Contenido. Muestra una ventana de ayuda de la forma de cómo utilizar el sistema.

Todos los módulos y catálogos del Sistema Administrador se componen por las mismas interfaces, lo que los hace diferentes es la forma en que fueron configurados en el “Generador de Código”, y los elementos que los integran, además de los permisos con que cuenta el usuario cuando accede al sistema. Dentro del sistema existe un módulo especial llamado “Usuarios”, este módulo, con base en sus interfaces, permite administrar los privilegios y permisos con que cuenta cada usuario. Para acceder a este módulo se debe contar con perfil de administrador.

### **Módulo de Usuarios**

El módulo de Usuarios posee un menú de iconos con todas las opciones para administrar la información relacionada con los usuarios del sistema. Las opciones de menú que presenta el módulo de usuarios son:

- Consulta. Esta interfaz es la principal, se despliega por defecto al acceder al módulo y contiene un menú con las opciones a las que el usuario tiene permiso de ejecutar.
- Buscar. Esta opción permite realizar la búsqueda de usuarios por los campos representativos de usuarios, como son: el nombre, login etc.
- Nuevo. Esta opción permite agregar nuevos usuarios, de acuerdo a las tareas a desarrollar se le asignarán los permisos dentro del sistema.
- Eliminar. Esta opción permite el borrado lógico de usuarios, deshabilitar los registros con base en un campo con valor booleano.
- Imprimir. Al ser seleccionada esta opción, y de acuerdo a lo que muestre la interfaz de consulta, abrirá la aplicación de “vista previa” del sistema operativo, para ser enviada a la impresora que esté conectada al equipo de cómputo.
- Filtrar. Esta opción permite la creación de filtros de datos, lo cual permite declarar criterios sobre los valores de los campos; el resultado de aplicar filtros se observa en la interfaz de consulta.
- Agrupar. La opción de agrupar permite arrastrar y conjuntar campos por columnas, permitiendo una mejor visibilidad y ordenamiento de los registros.



- Columnas. Esta opción despliega una ventana con los nombres descriptivos de todas las columnas, permitiendo elegir los campos que se deseen desplegar dentro de la interfaz de consulta.
- Cerrar. Cierra el módulo de usuarios.

Estas opciones que han sido descritas son genéricas para todos los módulos de acuerdo a los permisos y privilegios del usuario. Más adelante se detallará el funcionamiento de cada opción.

Este módulo es sumamente importante ya que incorpora la seguridad para el acceso a la información. Los módulos y catálogos restantes tienen un funcionamiento similar, incorporando las características definidas en el momento de su configuración.

## **Módulos y Catálogos**

El sistema administrador de información se compone de todos los módulos mencionados anteriormente en la descripción de menús. La operatividad de cada módulo es independiente de los demás, un usuario puede interactuar con uno o todos los módulos, teniendo en cada uno de ellos los mismos o distintos permisos.

Las interfaces que intervienen en los módulos y catálogos que componen el sistema son genéricas, es decir la funcionalidad que proporcionan es la misma para todos los módulos y catálogos, pero manejando información correspondiente a cada uno de ellos.

En la figura 5.18. se presenta un diagrama que describe la funcionalidad al acceder a un módulo o catálogo, esta acción abrirá la interfaz de consulta, presentando un menú de iconos, de acuerdo a los permisos que le fueron asignados al usuario, para ese módulo o catálogo en particular.

La opción módulos muestra los elementos a administrar de la base de datos, al seleccionar algún módulo se despliega la interfaz de consulta genérica, mostrando la información correspondiente con el módulo seleccionado. Esta interfaz es la más importante, ya que despliega el resultado de cualquier acción de acuerdo a la manipulación de la misma.

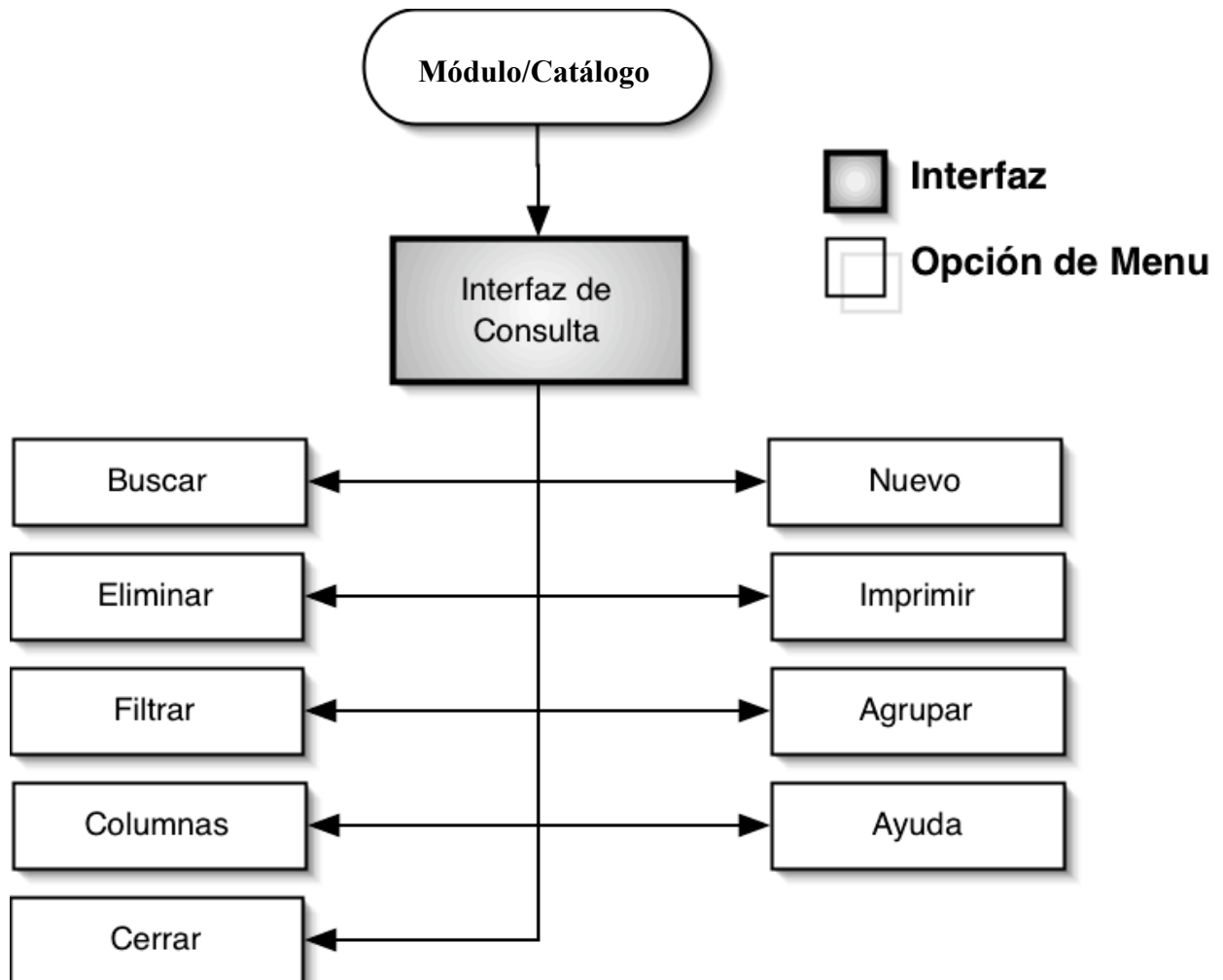


Figura 5.18. Inicio de Interfaces para módulos/catálogos.

En la figura 5.19. se muestra la interfaz de consulta para el módulo alumnos, podemos observar que contiene un menú formado por un grupo de iconos. Estos iconos proporcionan la manipulación de la información dentro de la interfaz consulta de alumnos, permitiendo realizar búsquedas, el alta de registros, el borrado de registros, el envío de la información que se muestra en la consulta a la impresora, creación de filtros para los datos que se muestran en la consulta, selección de columnas que aparecen dentro de la interfaz, agrupación de datos, ayuda y el botón para cerrar la interfaz.



121 Registro(s) Encontrado(s)

<input type="checkbox"/>	Número de cuenta	Nombre	Apellido Paterno	Apellido Materno
<input type="checkbox"/>	100092320673	LUIS ANTONIO	VAZQUEZ	RAMOS
<input type="checkbox"/>	102006000222	MIGUEL ANGEL	RAMREZ	LANDIN
<input type="checkbox"/>	102006000206	MA. DE LOS ANG...	VELASCO	ALVAREZ
<input type="checkbox"/>	130470000097	OYUKY	RAMREZ	BURCIAGA
<input type="checkbox"/>	680010000015	FERNANDO	GARCIA	MORENO
<input type="checkbox"/>	100001000288	GABRIELA	RADILLA	ROMANO
<input type="checkbox"/>	112002000175	ROBERTO	GOMEZ	GAITAN
<input type="checkbox"/>	127003000099	TERESA	SERVIN	MARTINEZ
<input type="checkbox"/>	100001000262	MARIA INES	DELGADO	GALEANA
<input type="checkbox"/>	102006000172	ALVARO ROLANDO	RIVERA	ARENAS
<input type="checkbox"/>	112002000159	HUMBERTO	GUTIERREZ	RAMREZ
<input type="checkbox"/>	127001000083	CARLOS	PACHECO	VAZQUEZ
<input type="checkbox"/>	112001000440	JUAN CARLOS	GOMEZ	SALDAÑA
<input type="checkbox"/>	102006000123	MARIA DEL ROCIO	DE LA MORA	GUZMAN
<input type="checkbox"/>	112001000024	MARIA DELA LUZ	HERNANDEZ	VILLALBA

Figura 5.19. Interfaz de Consulta.

La interfaz de consulta presenta los registros del módulo o catálogo seleccionado. Al costado izquierdo de cada registro se presenta un botón en forma de imagen, el cual permite acceder a la interfaz de modificación, con los datos correspondientes al registro. También se presenta una caja de selección, con la cual se pueden elegir uno o varios registros, con el propósito de ejecutar la opción de borrado lógico de los registros, la cual estará presente siempre que el usuario tenga este privilegio dentro del módulo.

La imagen que se muestra en la figura 5.20. muestra la interfaz de modificación para el módulo de alumnos, esta interfaz sólo permite sobrescribir valores de las propiedades que sean editables.

*Figura 5.20. Interfaz de modificación.*

Como se observa en la anterior figura, esta interfaz cuenta con un menú con las siguientes opciones:

- Guardar. Permite guardar los cambios y cerrar la interfaz, volviendo a la interfaz de consulta.
- Guardar y Nuevo. Realiza la acción de guardar el registro, y presenta la forma nuevamente para agregar un nuevo registro.
- Ayuda. Presenta la ayuda sobre el funcionamiento de la interfaz.
- Cerrar. Cierra la interfaz sin realizar cambios.

Al cerrar esta interfaz regresa automáticamente a la interfaz de consulta, mostrando los cambios realizados en el registro.

Como se mencionó anteriormente, al seleccionar los registros deseados, por medio de las cajas de selección (checkbox) de la interfaz de consulta y presionar la opción de borrar, se realiza el borrado lógico de los registros seleccionados. Como

ejemplo en la figura 5.21. se muestra la consulta de alumnos, eliminando algunos registros.

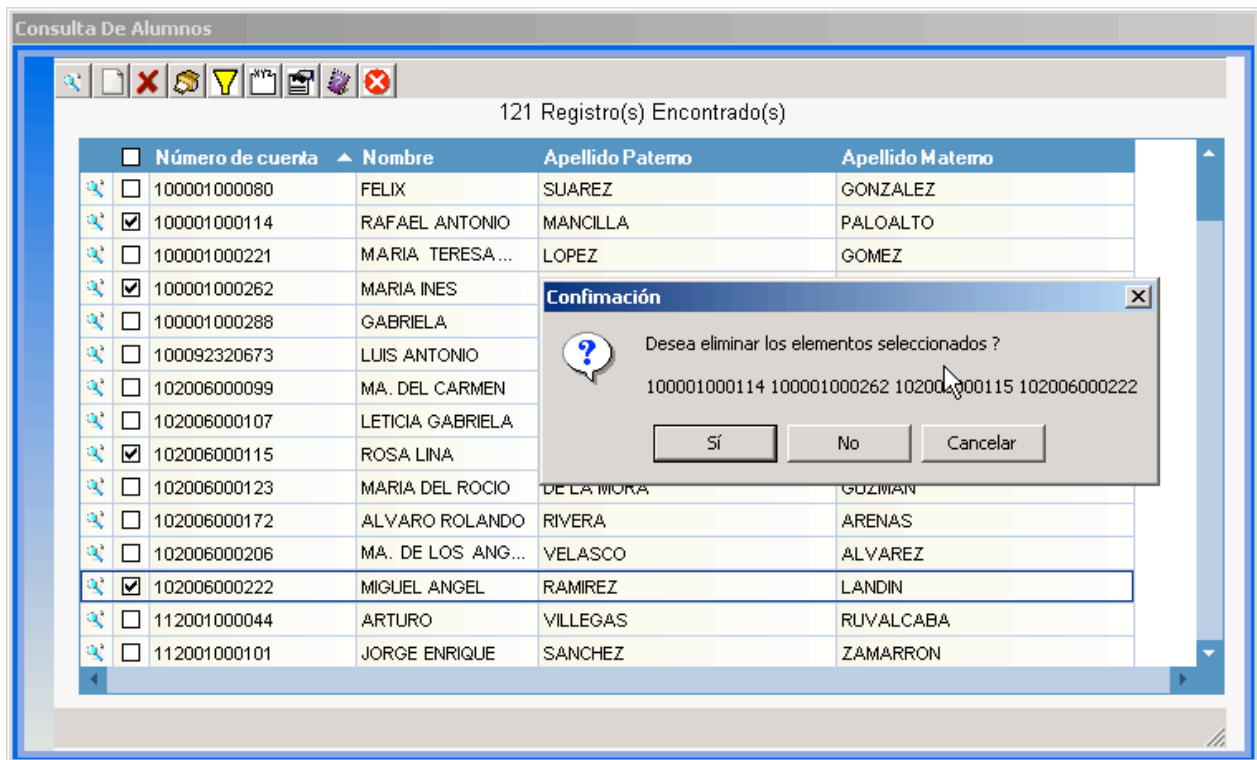


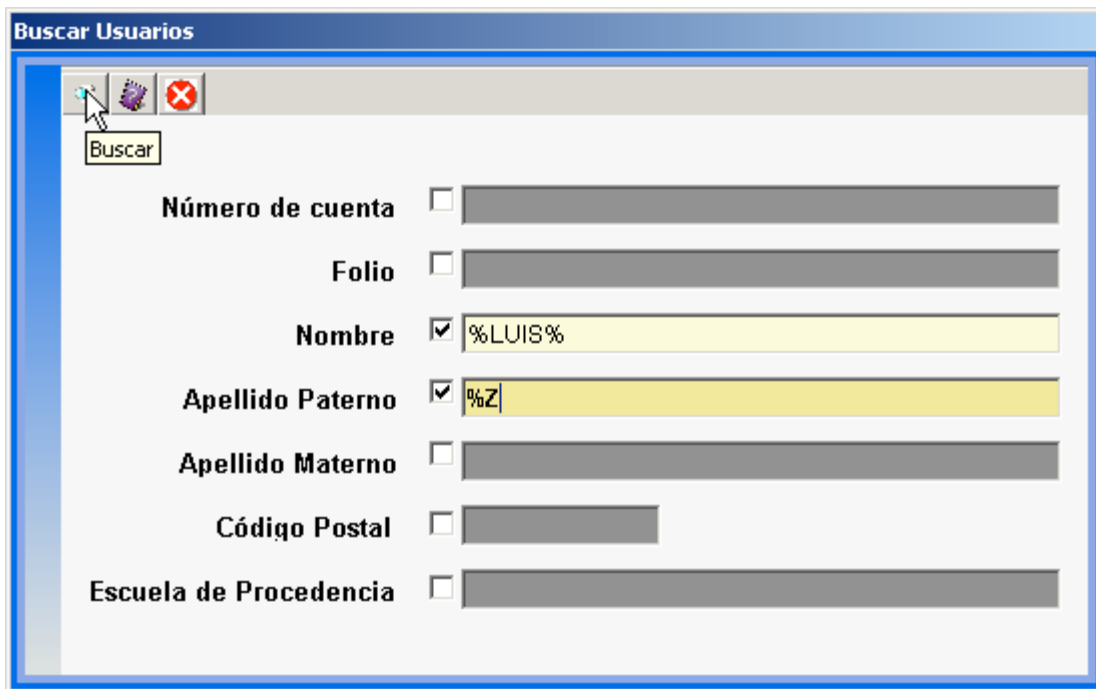
Figura 5.21. Eliminación de registros.

Al presionar sobre la opción de borrar, envía un mensaje para confirmar los registros que se desean eliminar, desde la cual podemos aceptar o cancelar, la acción.

Dentro del menú de la interfaz de consulta existe la opción buscar, está opción permite realizar búsquedas a través de una interfaz que despliega una forma con las propiedades elegidas, a partir de su configuración, para ser propiedades de búsqueda de patrones dentro de todos los registros.

La interfaz de búsqueda de alumnos se muestra en la figura 5.22, la cual contiene en su menú las siguientes opciones:

- Buscar. Ejecuta la búsqueda de acuerdo a los campos seleccionados y al posible valor en caso de ser numérico, o contenido si se trata de una propiedad con valor alfanumérico; una vez realizada esta acción muestra el resultado en la interfaz de consulta.
- Ayuda. Presenta una explicación acerca de cómo opera la interfaz.
- Cerrar. Cierra la interfaz y regresa a la interfaz de consulta.



Buscar Usuarios

Buscar

Número de cuenta

Folio

Nombre  %LUIS%

Apellido Paterno  %Z

Apellido Materno

Código Postal

Escuela de Procedencia

Figura 5.22. Interfaz de búsqueda.

En caso de no existir algún registro con las acotaciones definidas no regresará ningún registro, en caso de que no se definan patrones de búsqueda, regresará todos los registros.

La opción Nuevo permite agregar un nuevo registro, despliega una interfaz con todas las propiedades de acuerdo al tipo de dato que debe de contener, ya sea que se trate de un campo alfanumérico con una longitud máxima y mínima de caracteres, de un valor numérico con un valor máximo y mínimo o alguna de las dos reglas, o presente un catálogo en forma de lista, o una lista formada por botones de selección (vea campo sexo en la figura 5.23, como ejemplo). Así, en la parte superior se presenta un menú con las siguientes opciones:

- Guardar. Permite guardar el nuevo registro y cierra la interfaz, volviendo a la interfaz de consulta.
- Guardar y Nuevo. Realiza la acción de guardar el registro, y presenta la forma nuevamente para agregar un nuevo registro.
- Ayuda. Presenta la ayuda sobre el funcionamiento de la interfaz.
- Cerrar. Cierra la interfaz sin realizar cambios.

La figura 5.23 muestra la interfaz de modificación, en este caso sirve para ejemplificar el alta de un Alumno.

The screenshot displays the 'Modificar Alumnos' (Modify Students) interface. It is divided into several sections:

- Personal Information:** Includes fields for 'Estado Civil' (set to 'Soltero'), 'Número de cuenta', 'NIP', 'CURP', 'Folio', 'Nombre', 'Apellido Paterno', 'Apellido Materno', 'Fecha de Nacimiento' (07/07/2003), 'Calle y Número', and 'Código Postal'.
- Location and Demographics:** Includes 'Estado', 'Ciudad', 'Delegación ó Municipio', 'Colonia', 'Sexo' (radio buttons for Masculino and Femenino), 'Fotografía' (with a 'Fotografía' button), 'Edad', 'Nacionalidad' (radio buttons for Nacional, Extranjero, and Otro), 'Fecha de Inscripción' (07/07/2003), 'Escuela de Procedencia', and 'Código de Barras'.
- Academic Records:** A table titled 'Carreras' with buttons for 'Columnas', 'Agregar', and 'Eliminar'. Below it are sections for 'Documentos Recibidos', 'Exámenes', 'Padres', and 'Status'.
- Document Management:** A table titled 'Documentos Entregados' with buttons for 'Columnas', 'Agregar', and 'Eliminar'. Below it are sections for 'Emails', 'Medicina', 'Pagos', and 'Teléfonos'.

5.23. Interfaz de Modificación.

Esta interfaz tiene la funcionalidad tanto de alta como de modificación de los datos de Alumnos, y se presenta en un modo o en otro dependiendo de la funcionalidad a realizar.

La opción imprimir permite enviar el resultado, que aparece en la interfaz de consulta, a la aplicación de vista previa del sistema operativo, para que pueda ser impresa. En la figura 5.24. se presenta el resultado al aplicar esta operación sobre el módulo de consulta.



Figura 5.24. Vista previa para enviar a impresora.

La opción de filtrar permite establecer criterios de búsqueda estableciendo rangos y términos, que caracterizan al valor desplegado en una determinada columna del grid de consulta, con el propósito de mostrar los registros que en determinado momento le interesen al usuario. En la figura 5.25. se muestra la interfaz que permite establecer la propiedad, el criterio y el patrón para el filtro, ya sea para un campo tipo texto o numérico.

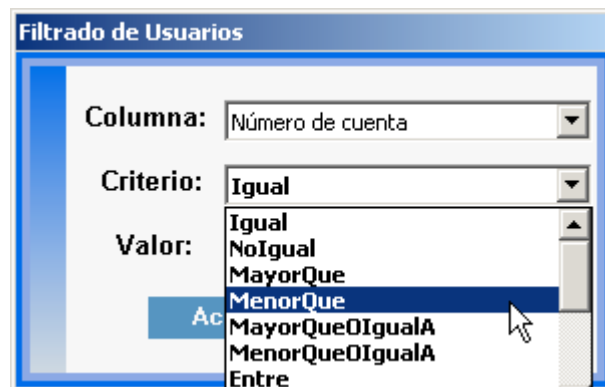


Figura 5.25. Definición del filtro.

Después de aplicar los criterios de filtrado, se presenta la interfaz de consulta que se muestra en la figura 5.26.



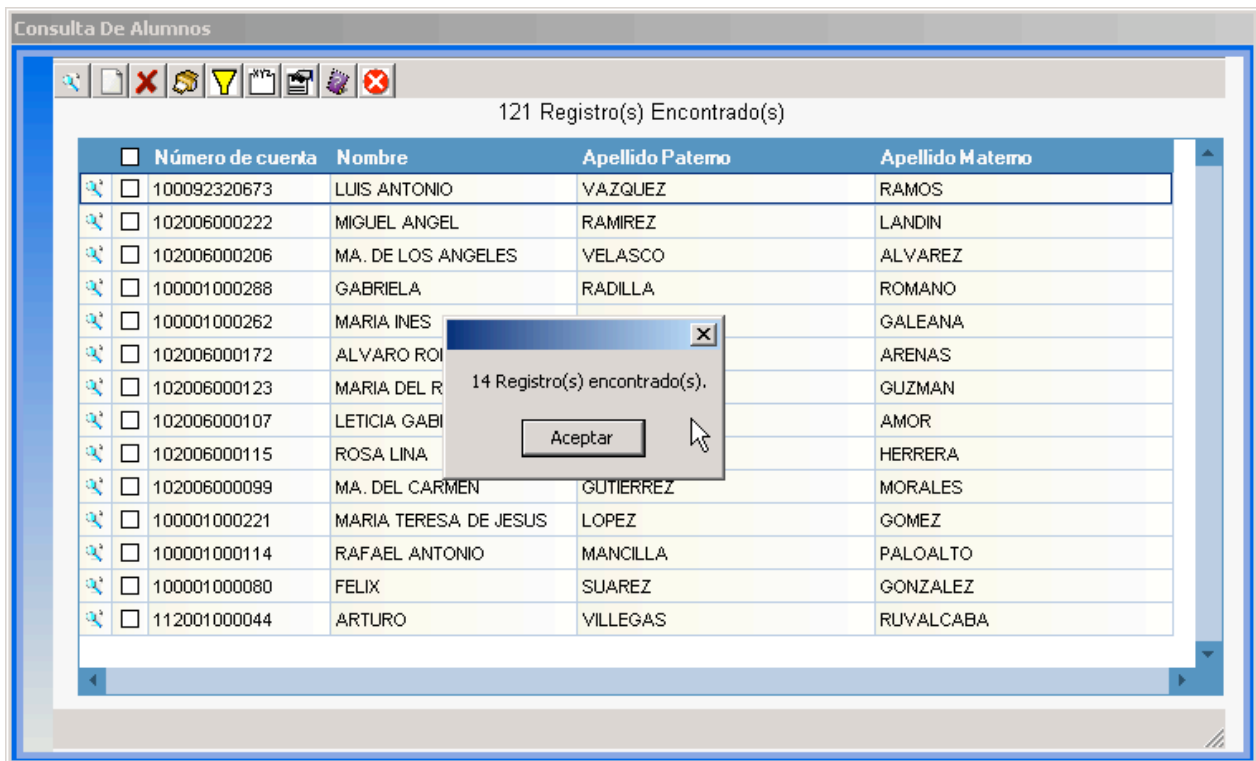


Figura 5.26. Resultado de aplicar un filtro sobre alguna columna.

El resultado del filtro abre una ventana con el mensaje indicando la cantidad de registros que cumplen con la condición establecida en el filtro

La opción agrupar, al ser seleccionada dentro del menú de consulta, despliega en la parte superior de la interfaz de consulta una barra como se puede observar en la figura 5.27. En esta sección de la interfaz se puede arrastrar cualquier etiqueta de las columnas que se visualizan en ella. Al realizar esta acción, la información de los registros se irá agrupando de acuerdo al orden en que las etiquetas de las columnas hayan sido arrastradas a la parte superior.

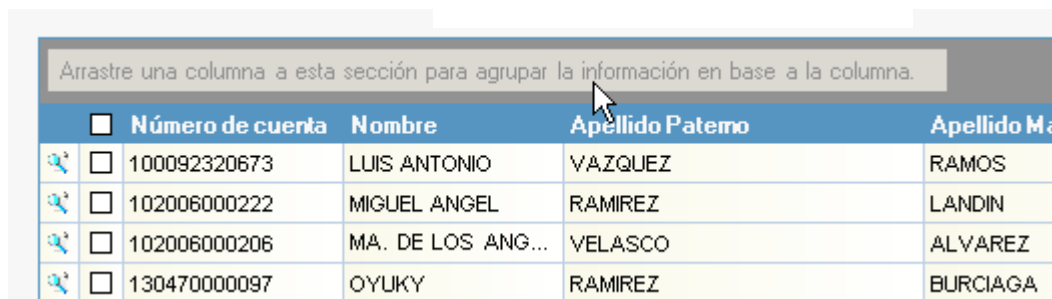


Figura 5.27. Habilitando la funcionalidad agrupar.

La figura 5.28. muestra el resultado de arrastrar en la interfaz alumnos la etiqueta de la columna apellido paterno. Se puede observar en la figura la forma en como la opción de agrupar modifica la forma en que la interfaz de consulta presenta la información del módulo, agrupando la información en columnas.



The screenshot shows a window titled "Consulta De Alumnos" with a toolbar and a status bar indicating "121 Registro(s) Encontrado(s)". A dropdown menu is open for "Apellido Paterno", showing a list of names grouped by their paternal surnames. The table below displays the data with columns for account number, name, paternal surname, and maternal surname.

<input type="checkbox"/>	Número de cuenta	Nombre	Apellido Paterno	Apellido Materno
<input type="checkbox"/>	Apellido Paterno: GARCIA			
<input type="checkbox"/>	Apellido Paterno: GARISDAIN			
<input type="checkbox"/>	Apellido Paterno: GODINA			
<input type="checkbox"/>	Apellido Paterno: GOMEZ			
<input type="checkbox"/>	112002000175	ROBERTO	GOMEZ	GAITAN
<input type="checkbox"/>	112001000440	JUAN CARLOS	GOMEZ	SALDAÑA
<input type="checkbox"/>	Apellido Paterno: GONZALEZ			
<input type="checkbox"/>	240350000206	TERESA	GONZALEZ	CORTEZ
<input type="checkbox"/>	305600000650	RUBY EDITH	GONZALEZ	BORRAS
<input type="checkbox"/>	120060000516	HOMERO	GONZALEZ	MEDRANO
<input type="checkbox"/>	112002000100	HILDEBERTO	GONZALEZ	ENRIQUEZ
<input type="checkbox"/>	Apellido Paterno: GUERRERO			
<input type="checkbox"/>	802500001283	JUAN DIEGO EDSON...	GUERRERO	COLONGA
<input type="checkbox"/>	Apellido Paterno: GUTIERREZ			
<input type="checkbox"/>	102006000099	MA. DEL CARMEN	GUTIERREZ	MORALES

Figura 5.28. Agrupación de Columnas.

Al seleccionar la opción columnas, de la interfaz de consulta, ésta despliega una interfaz con las propiedades pertenecientes al módulo que en ese instante no aparecen en la interfaz de consulta, ver figura 5.29. Interactuando la interfaz de columnas y de consulta se puede agregar u ocultar cualquier columna de ésta, a través de eventos del mouse o *trakpad* (tomar, arrastrar y soltar) entre estas interfaces. En la interfaz de columnas puede ser seleccionada una de varias de las columnas del módulo, por medio del mouse, para que se muestren en la interfaz de consulta. Esto aumentará o disminuirá el número de columnas que se visualizan dentro de la interfaz de consulta.

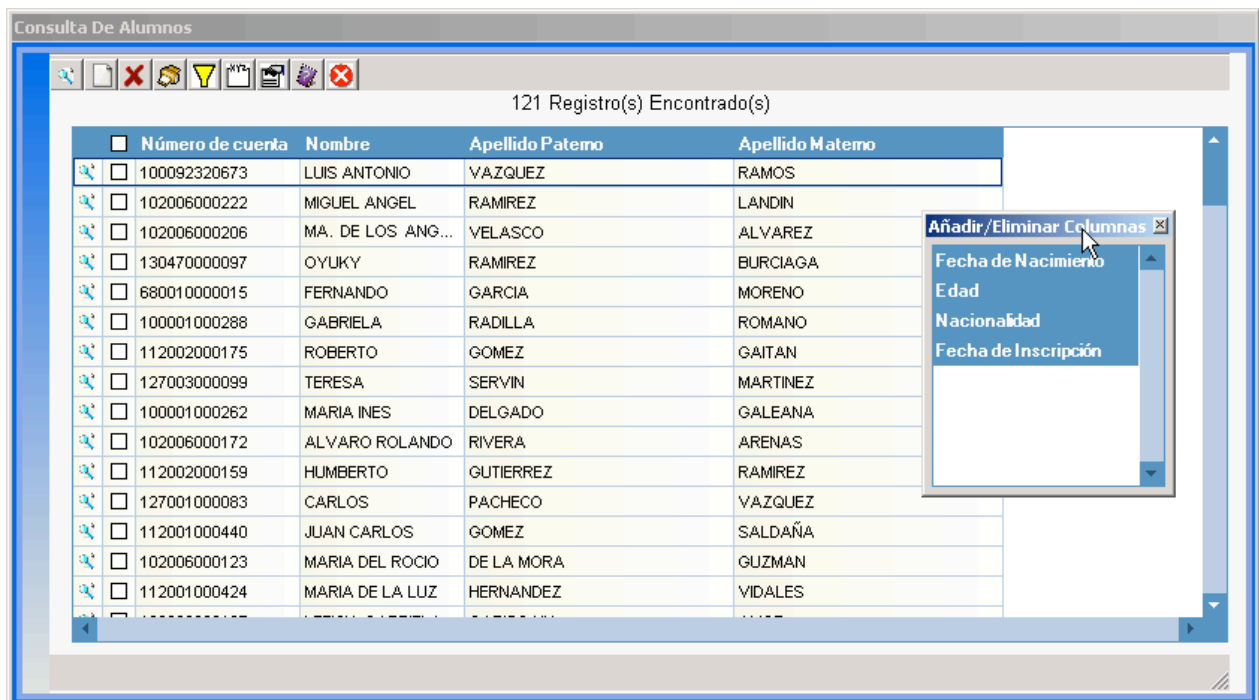


Figura 5.29. Agregando/Eliminando columnas en Consulta.

Además esta funcionalidad permite establecer el orden en que aparecen las columnas dentro de la interfaz de consulta.

La opción ayuda, en el menú de consulta, presenta la descripción de cada opción del menú así como el manual de usuario para el uso correcto del sistema.

La opción cerrar, cierra la interfaz de consulta y ubica la funcionalidad en la interfaz principal, donde se puede acceder a otro módulo o catálogo, salir de sesión, acceder al módulo de ayuda general y cerrar el sistema.

Hasta este punto se han presentado las interfaces del sistema ejemplificando con la entidad alumno. En el siguiente apartado se hablará sobre las pruebas a las que fue sometido el sistema, su instalación, puesta a punto y liberación.

### 5.3. Pruebas e Instalación del Sistema

Una de las etapas principales dentro del desarrollo de un sistema es la etapa de pruebas, las cuales consisten en un conjunto de actividades necesarias para encontrar fallas y proporcionar la confianza de que el sistema se ajusta a los requisitos técnicos establecidos y a los requerimientos del cliente, cumpliendo con altos estándares de calidad.

Las pruebas permiten:

- Verificar los requerimientos funcionales y estructurales para establecerlos como fundamento para la realización de las pruebas.
- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera correcta y consistente.
- Buscar y registrar fallas o defectos asociados a los requerimientos establecidos.

Existen varios tipos de pruebas, las cuales pueden ir desde el análisis del desempeño de un módulo del sistema y las estructuras lógicas que lo conforman, hasta las pruebas del sistema en su conjunto.

## **Prueba de Unidad**

Las pruebas de unidad centran el proceso de verificación en la menor unidad del diseño del software: el componente de software o módulo, antes de la integración de la unidad del programa en un sistema más grande.

Una unidad de programa debe ser lo suficientemente pequeña como para que el programador que la desarrolló pueda probarla con minuciosidad, esta prueba debe ser mucho más minuciosa que el examen al que se someterá cuando la unidad se integre en el producto de software en desarrollo, por otra parte, debe resaltarse que es sumamente importante que el programador la lleve a cabo y que no se deje de realizarla por omisión.

En las pruebas de unidad, el código fuente debe cumplir satisfactoriamente las pruebas de función, de desempeño, de tensión y estructurales. Las dos primeras se basan en las especificaciones de los requisitos y se diseñan para demostrar que el sistema satisface los mismos.

Dichos casos de prueba funcionales especifican las condiciones operativas comunes, valores de entrada y resultados comunes esperados. Las pruebas de función también prueban el comportamiento exactamente dentro, sobre y más allá de las fronteras funcionales.

Las pruebas de desempeño se proyectan para verificar el tiempo de respuesta bajo cargas variables, porcentaje del tiempo de ejecución empleado en varios segmentos del programa, el rendimiento y las tasas de tráfico en los canales de datos y los enlaces de comunicación.

Las pruebas de tensión se diseñan para sobrecargar un sistema de varias formas. Entre los ejemplos de las pruebas de tensión se encuentran el intentar asignar

más del número máximo de terminales permitidas, procesar más del número permitido de identificadores o desconectar un enlace de comunicación.

Las pruebas de estructura se relacionan con el examen de la lógica interna de procesamiento de un sistema de software. Las rutinas particulares llamadas y las rutas lógicas recorridas a lo largo de las rutinas son los objetos importantes. La meta de las pruebas de estructura es recorrer en el sistema un número específico de rutas, a través de cada rutina, de modo que se establezca la profundidad de las pruebas. Este tipo de pruebas a menudo se utilizan como una forma de aumentar las pruebas funcionales, de desempeño y de tensión, por lo que no pueden diseñarse sino hasta que el sistema fue implantado y sometido al plan de pruebas definido.

Cada caso de prueba debe proporcionar la siguiente información:

- a) El tipo de prueba aplicado (función, desempeño, tensión, estructural).
- b) Configuración o requisitos preliminares de la máquina donde se realizará la prueba, o bien, todos los requisitos de hardware y/o software adicional, así como en su caso acceso a bases de datos o firmas especiales para ingresar a ambientes de prueba.
- c) Suposiciones efectuadas en la prueba.
- d) Requisitos de la prueba (condiciones previas).
- e) Estímulo exacto de la prueba.
- f) Resultado esperado.

La estrategia de la prueba es un producto de trabajo importante del proceso de diseño y, como todos los productos de trabajo, se debe desarrollar de una forma sistemática, con la particularidad de que su evaluación debe contar con un visto bueno del usuario final.

## **Prueba de Integración**

La prueba de integración es una técnica sistemática para construir la estructura del programa, mientras que al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es seleccionar los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

La integración del programa para la realización de esta prueba debe realizarse de forma incremental, o sea el programa se construye y se prueba en pequeños segmentos en que los errores son más fáciles de corregir, es más probable que se

puedan probar completamente las interfaces y se puedan aplicar un enfoque de prueba sistemática. Entre las estrategias de integración se encuentran las siguientes:

### *Integración Descendente*

La prueba de Integración descendente es un planteamiento incremental a la construcción de la estructura de programas. Se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando por el módulo de control principal. Los módulos subordinados al módulo de control principal se van incorporando en la estructura. Esta integración se realiza en una serie de cinco pasos:

1. Se usa el módulo de control principal como controlador de la prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración elegido( ya sea en profundidad o en anchura) se van sustituyendo por los módulos reales.
3. Se llevan a cabo pruebas cada vez que se integra un nuevo módulo.
4. Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo del módulo real.
5. Se hace la prueba de regresión para asegurarse de que no sean introducidos errores nuevos.

El proceso continúa desde el paso 2 hasta que se haya construido la estructura del programa entero.

### *Integración ascendente*

La prueba de Integración ascendente, como su nombre lo indica, empieza la construcción y la prueba con los módulos atómicos (es decir, módulos de los niveles más bajos de la estructura del programa). Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos.

Se puede implementar una estrategia de integración ascendente mediante los siguientes pasos:

1. Se combinan los módulos de bajo nivel en grupos (a veces denominados construcciones) que realicen una subfunción específica del software.
2. Se escribe un controlador (un programa de control de prueba) para coordinar la entrada y la salida de los casos de prueba.
3. Se prueba el grupo.

4. Se eliminan los controladores y se combinan los grupos moviéndose hacia arriba por la estructura del programa.

Las pruebas ascendentes tienen la desventaja de necesitar escribir y depurar ambientes de prueba para los módulos y subsistemas, además, el nivel de complejidad adquirido al combinar los módulos y subsistemas en unidades cada vez más grandes. El caso extremo de complejidad resulta cuando cada uno de los módulos se liga y ejecuta en una sola corrida de integración y el principal problema en este punto, será el aislar las fuentes de los errores.

Aunque pudiera parecer que la integración descendente es preferible, se presentan muchas ocasiones en las que es imposible apegarse a una estrategia de integración y pruebas estrictamente descendente. Por ejemplo, puede ser difícil encontrar datos de entrada del nivel más alto que ejerciten un módulo de más bajo nivel en la forma deseada. También puede ser demasiado caro correr el sistema en desarrollo como un ambiente de prueba para las nuevas rutinas. A menudo se puede ahorrar tiempo de forma significativa si los subsistemas se prueban por separado antes de insertarlos en la estructura descendente en desarrollo.

La selección de una estrategia de integración depende de las características del software y, a veces, de la planificación del proyecto. En general, el mejor compromiso puede ser un enfoque combinado (a veces denominado prueba sandwich), que use la descendente para los niveles superiores de la estructura del programa, junto con la ascendente para los niveles subordinados.

## **Prueba de Validación**

Las pruebas de validación o aceptación implican la planeación y la ejecución de pruebas funcionales, de desempeño y de tensión para demostrar que el sistema implantado satisface sus requisitos. Con frecuencia las pruebas que se corren son desarrolladas por el equipo de control de calidad, con el visto bueno del cliente, quien a su vez en algunas ocasiones aplica sus propios casos de prueba y revisa los resultados obtenidos con los casos de prueba que previamente autorizó.

Además de las pruebas funcionales y de desempeño, las pruebas de tensión se llevan a cabo con el fin de establecer las limitaciones del sistema.

Por lo común, las pruebas de aceptación incorporan casos de prueba desarrollados durante las pruebas unitarias y de aceptación. Se añaden casos de prueba adicionales para lograr el nivel deseado de las pruebas funcional, de desempeño y de tensión del sistema completo. Entre las herramientas de especial importancia durante las pruebas de aceptación se hallan un analizador de cobertura de la prueba y otro de tiempos, además un verificador de los estándares de codificación.

Un analizador de la cobertura de la prueba registra las rutas de control seguidas por cada caso de prueba. El registro acumulado sirve para fijar la extensión de la cobertura de la prueba obtenida durante la prueba de aceptación. En los grandes sistemas, el analizador de la cobertura puede registrar sólo las rutinas llamadas y no las proposiciones individuales ejecutadas.

Un analizador de tiempos informa el tiempo empleado en varias regiones del código fuente bajo diferentes casos de prueba.

Finalmente, el analizador de estándares de codificación revisará los estándares que pueden estar relacionados con el proyecto, con el cliente, o con el lenguaje.

En este punto es necesario destacar la importancia de las pruebas de integración y validación, ya que con ambos procesos se garantiza el correcto funcionamiento del producto de programación que será entregado al cliente, así como la satisfacción de todos los requerimientos que el cliente realizó; por otra parte, debe señalarse que ningún producto de programación puede ser entregado al cliente si no se obtuvo su visto bueno sobre las pruebas de aceptación.

En resumen los principales beneficios de la aplicación de las técnicas de verificación y validación son:

1. Detección y corrección oportuna de los errores del sistema, en cada etapa del proceso de pruebas.
2. Establecimiento de estándares para el reporte y desarrollo de las adecuaciones necesarias al sistema.
3. Registro y seguimiento a los cambios de alcance solicitados por el cliente.
4. Garantía de la satisfacción de los requerimientos del cliente, así como de los requisitos tecnológicos esperados del sistema, al contar con un equipo de pruebas diferente al de desarrollo.
5. Identificación de los conocimientos que deberán reforzarse en los elementos del equipo, al registrar los módulos con más problemas, el tipo de errores presentados con mayor frecuencia, así como quien los generó y el tiempo que tomó resolverlos.
6. Generación de la documentación que permitirá realizar un análisis sobre las desviaciones en costos y tiempo.
7. Involucramiento del cliente en la etapa final de las pruebas, lo cual garantiza su aprobación y conocimiento, antes de ser operado por el usuario final.
8. Generación de los manuales de capacitación para el cliente, así como de los manuales técnicos.



9. Adquisición de un mayor conocimiento del negocio del cliente por medio de la interacción entre el equipo de pruebas y el cliente.

Identificación y documentación de las posibles mejoras del sistema, al analizar los cambios en el alcance que no podrán ser aplicados en la presente etapa del desarrollo del producto de software, pero que se podrán implementar posteriormente.

## Prueba del Sistema

Las pruebas del sistema se relacionan con base en las interfaces, la lógica de decisión, el flujo del control, los procedimientos de recuperación, la eficiencia global, la capacidad y las características de tiempo de respuesta del sistema entero. Se requiere una escrupulosa planeación de las pruebas para determinar la extensión y la naturaleza de las pruebas del sistema que se van a realizar, y establecer los criterios para la evaluación de los resultados.

La prueba del sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

*Prueba de recuperación.* La prueba de recuperación es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente.

*Prueba de seguridad.* La prueba de seguridad intenta verificar que los mecanismos, de protección incorporados en el sistema la protegerán, de hecho, de accesos impropios.

*Prueba de Resistencia.* La prueba de resistencia ejecuta un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales.

*Prueba de Rendimiento.* La prueba de rendimiento está diseñada para probar el rendimiento del software en tiempo de ejecución, dentro del contexto de un sistema integrado.

Para la realización de esta etapa se debe contar con un ambiente de pruebas adecuado, en nuestro caso consistió en la instalación del sistema, generación de datos de prueba, identificación de las herramientas (para nuestro trabajo la base de datos SQL server 2000 y el programa instalador del Sistema Administrador generado por herramientas de .NET).

Además se deben tomar en cuenta para la realización de pruebas aspectos como los puntos de control, el procedimiento a seguir, la forma de de evaluación y conclusión de las mismas.

### 5.3.1. Pruebas Aplicadas

En la aplicación de pruebas a nuestro sistema se establecieron los puntos de control y de aprobación entre las interfaces del mismo, además de probar todas las opciones que brinda el sistema con datos válidos y no válidos. Se verificó que si la opción seleccionada a su vez contaba con varias opciones, éstas fueron verificadas. El punto de control principal para el alta, modificación y alteración de la información en cualquier módulo o catálogo fue la interfaz de consulta, ya que su avanzado diseño permite la comprobación de cualquier operación. Se probaron todas las opciones del menú principal, que la referencia fuera la correcta y que además estuviera de acuerdo al rol asignado al usuario dentro del módulo de usuarios.

Las pruebas que se aplicaron al Sistema Administrador de Información para una Institución Educativa fueron las siguientes:

- De Unidad.
- De Integración.
- De Validación.
- De Sistema.

#### Prueba de Unidad

Las pruebas de unidad se aplicaron a cada uno de módulos (Usuarios, alumnos, Padres o Tutores, Pagos, Planes de Estudio y Profesores) y catálogos (Carreras, Conceptos de Pago, Especialidades, Estado Civil, Grados Académicos, Medicina General, Motivos de Inasistencia, status, Tipos de Examen, Tipos de Documentos, Tipos de Profesores y Tipos de Teléfono) de manera independiente, los cuales integran al sistema administrador de información, para descubrir errores de lógica y de funcionamiento, las pruebas comprendieron lo siguiente:

- Ejecución de las decisiones lógicas.
- Ejecución de los caminos independientes de cada módulo.
- Ejecución de aplicaciones en sus límites operacionales.

Estas pruebas fueron realizadas por los desarrolladores y los analistas del sistema, en el cual se garantizó que se ejecutaran de forma correcta por lo menos todos los caminos independientes de cada módulo. Se probaron todas las funcionalidades con que cuenta cada módulo, validando datos tanto de entrada como salida, obteniendo los resultados esperados, este proceso resultó exitoso, demostró la eficiencia del

Generador de Código al construir nuestro sistema. Se verificó que al introducir datos erróneos el sistema despliega mensajes de error.

## **Prueba de Integración**

Esta prueba consistió en probar la funcionalidad de todo el sistema en conjunto, desde la comprobación del funcionamiento del menú principal y submenús hasta la comprobación de resultados, al introducir datos y realizar peticiones al sistema, y comprobación de seguridad (creación de usuarios con cierto rol), aplicando la estrategia de integración descendente.

El sistema se comportó muy bien en este aspecto, no se detectó ningún problema grave, solamente errores en etiquetas y descripciones de funcionalidad que no eran muy claros, fueron necesarias mínimas correcciones.

Dichos casos de funcionalidad especifican las condiciones operativas comunes, valores de entrada y resultados comunes esperados. Las pruebas de integración también prueban el comportamiento exactamente dentro y sobre las fronteras funcionales, lo cual valida los requerimientos de la organización, y pretenden descubrir errores cometidos en la construcción del código a partir del Generador de Código.

## **Prueba de Validación**

En la ejecución de esta prueba se invitó a usuarios a probar el sistema para observar el grado de aceptación y recavar la información que no fue contemplada en el análisis de requerimientos, y sea tomada en cuenta como parte en el mantenimiento del sistema.

En este caso se probó con diferentes tipos de usuarios, de acuerdo al área y responsabilidad en el uso del sistema. Se obtuvieron buenos comentarios por la forma de acceso y la asignación de permisos (módulo de acceso y creación de usuarios). El sistema operó sin problema alguno, en esta prueba se obtuvieron las aprobaciones para llevar el sistema a producción.

## **Prueba del Sistema**

Se comprobó la funcionalidad de acuerdo a los requerimientos del sistema; se realizó la comprobación de cada interfaz, con base en las opciones de operatividad con que cuenta. Se validó que los usuarios definidos sólo podían acceder y ejecutar acciones de acuerdo al grupo y permisos asignados, además de verificar todos los casos al introducir datos válidos e inválidos, y que los resultados correspondieran con lo

especificado en los requerimientos, las pruebas de funcionalidad abarcaron todos los módulos, incluyendo el de usuarios, y catálogos.

### **5.3.2. Instalación**

En esta fase se llevó acabo la instalación del nuevo sistema en ambiente de producción. Para esto la institución ya contaba con la infraestructura tecnológica necesaria para el correcto funcionamiento del sistema, ya que en la etapa de análisis y diseño se tomaron en cuenta estos aspectos,

La fase o etapa de instalación constó de la realización de las siguientes actividades:

1. Instalar el sistema en el servidor. Esto consistió en instalar SQL Server 2000 en el servidor, el cual cuenta con las siguientes características:
  - Procesador Intel Pentium 4 a 2.2 GHz
  - 512 MB memoria SDRAM DDR
  - Disco Duro de 60 GB
  - Unidad Combo DVD/CD-RW
  - Tarjeta de Red Ethernet 10/100/1000
  - Sistema Operativo Windows XP Profesional
  
2. Transferencia masiva de información. La información contenida en archivos, una vez dado el formato necesario, y con base en operaciones y macros de Excel, se transfirió a la base de datos del sistema, llamada "SAIE" que es la abreviación del nombre del sistema (Sistema Administrador para una Institución Educativa). Esto fue un proceso difícil debido a la duplicidad de información y a falta de información en registros que en el sistema son obligatorios.
  
3. Instalar el sistema en los equipos clientes. En este punto se procedió a instalar el sistema en las máquinas cliente, que como parte de un esquema nuevo e independiente, se instalaron en equipos con Windows 2000 o versión posteriores. Al realizar la instalación se solicitó la presencia del jefe de sistemas y personal de soporte para que aprendieran la manera de instalación y comprobación del buen funcionamiento. Al mismo tiempo se solicitó se realizarán la revisión general a fin de poner a punto la operación del sistema.

Una vez terminada la instalación se procedió a la capacitación del personal, mostrando el correcto uso que se le debe dar a la herramienta, considerando aspectos importantes como el modo de acceso al sistema, utilización del menú principal de la aplicación, descripción de los mensajes de advertencia y error, y la explicación de cada interfaz del sistema.

Se respondieron a dudas de tipo operativo de posibles casos especiales que pudieran presentarse. Además se dio una capacitación especial al responsable de la administración de usuarios, ya que es el usuario que debe conocer al cien por ciento el sistema, se hizo énfasis en aspectos relevantes como el alta y la asignación de permisos a usuarios.

Realizado lo anterior se procedió deslindar nuestra responsabilidad del sistema y que ésta fuera asumida por el departamento de sistemas, procediendo a la liberación del mismo.

Como resumen del capítulo podemos mencionar que se han mostrado a detalle todos los pasos necesarios para la creación del Sistema Administrador de Información, desde la configuración del Generador de Código, hasta la configuración de cada una de las entidades que corresponden a la Institución Educativa. De la misma manera hemos ejemplificado el uso de cada una de las interfaces que se presentan en el sistema, tomando como ejemplo el módulo de Alumnos y presentando una amplia variedad de casos y mensajes que se presentan durante la interacción con la aplicación. También se mostró una gran variedad de pruebas a las que fue sometido el sistema para con ello garantizar tanto el cumplimiento a las expectativas de la Institución Educativa, así como un alto grado de confianza en los resultados obtenidos.

En la siguiente sección presentamos a manera de resumen los resultados obtenidos a lo largo de este trabajo, así como los objetivos cumplidos, las conclusiones y los puntos mas destacados que sirvieron para terminar este proyecto con éxito.

# Índice de Figuras

## Capítulo 1 - Generalidades

Figura 1.1. Esquema. de componentes dentro de la plataforma .NET Framework .....10

## Capítulo 2 - Análisis del Sistema

Figura 2.1. División de Direcciones y Coordinaciones .....18

Figura 2.2. Procedimiento para inscribir a los alumnos de nuevo ingreso .....24

Figura 2.3. Procedimiento de reinscripción de alumnos regulares .....30

Figura 2.4. Procedimiento de inscripción para alumnos irregulares .....33

Figura 2.5. Procedimiento para integrar el expediente físico del personal docente .....37

Figura 2.6. Procedimiento para diseñar y rediseñar planes de estudio .....39

Figura 2.7. Procedimiento para diseñar y rediseñar los programas de las materias ....44

## Capítulo 3 - Diseño del Sistema

Figura 3.1. Diagrama General del Sistema .....57

Figura 3.2. Módulo de Autenticación de Usuarios .....58

Figura 3.3. Diagrama del Módulo Principal .....59

Figura 3.4. Funcionalidad del Módulo de Consulta .....61

Figura 3.5. Modelo Entidad-Relación para la Autenticación de Usuarios .....63

Figura 3.6. Modelo E-R para el Módulo de Alumno .....64

Figura 3.7. Modelo E-R del Módulo Alumno-Padre o Tutor .....65

Figura 3.8. Módulo de Alumnos-Pagos .....66

Figura 3.9. Modelo E-R del Módulo de Profesores .....67

Figura 3.10. Modelo E-R del Módulo de Planes de Estudio .....68

Figura 3.11. Modelo E-R del Sistema Administrador de Información .....69

## Capítulo 4 - Generador de Código

Figura 4.1. Diagrama General del Proceso .....108

Figura 4.2. Diagrama de flujo del Generador de Código .....110

Figura 4.3. Modelo de datos del archivo de proyecto .....111

Figura 4.4. Nuevo Proyecto .....	120
Figura 4.5. Abrir un proyecto existente .....	121
Figura 4.6. Objeto CadenaDeConexion .....	122
Figura 4.7. Módulo de conexión a datos .....	123
Figura 4.8. Jerarquía de clases .....	124
Figura 4.9. Estructura de objeto Proyecto .....	126
Figura 4.10. Estructura de objeto Tablas .....	127
Figura 4.11. Estructura de objeto Campo .....	128
Figura 4.12. Estructura de objeto Relación .....	129
Figura 4.13. Estructura de objeto BotonDeOpcion .....	130
Figura 4.14. Estructura de objeto CajaDeElementos .....	130
Figura 4.15. Módulo de Configuración .....	131
Figura 4.16. Panel de configuración de proyecto .....	132
Figura 4.17. Panel de configuración de tablas .....	132
Figura 4.18. Panel de configuración de campos .....	133
Figura 4.19. Forma de selección estática .....	133
Figura 4.20. Forma de selección dinámica .....	134
Figura 4.21. Elementos del modelo de solución .....	135
Figura 4.22. Modelo jerárquico de la interfaz .....	137
Figura 4.23. Contenedores de objetos gráficos .....	138
Figura 4.24. Objetos de ejecución de funciones .....	138
Figura 4.25. Objetos de captura de caracteres .....	139
Figura 4.26. Objetos OptionButton de selección de uno entre varios .....	139
Figura 4.27. Objetos STComboBox de selección de uno entre varios .....	140
Figura 4.28. Objeto STContenedor.. .....	140
Figura 4.29. Objeto STGrid .....	141
Figura 4.30. Objetos de captura de datos lógicos .....	141
Figura 4.31. Objetos de captura de fechas .....	142
Figura 4.32. Pseudo código Método Consulta .....	144
Figura 4.33. Método Consulta .....	146
Figura 4.34. Pseudo código Método Alta .....	150
Figura 4.35. Método Alta .....	151
Figura 4.36. Pseudo código Método Baja .....	153
Figura 4.37. Método baja .....	153
Figura 4.38. Pseudo Código Método Modificación .....	159
Figura 4.39. Método Modificación .....	160

## Capítulo 5 - Diseño del Sistema

Figura 5.1. Interacción del Usuario .....	164
Figura 5.2. Generación del sistema administrador de Información .....	165
Figura 5.3. Configuración de la entidad proyecto .....	168
Figura 5.4. Configuración de la entidad alumnos .....	169
Figura 5.5. Configuración del campo Idalumno .....	171
Figura 5.6. Configuración del campo Número de cuenta .....	171
Figura 5.7. Configuración del campo sexo .....	172
Figura 5.8. Configuración del campo IDEstadoCivil .....	173
Figura 5.9. Generación de Código .....	174
Figura 5.10. Inicio del Sistema .....	175
Figura 5.11. Pantalla de inicio .....	176
Figura 5.12. Pantalla acceso al sistema .....	177
Figura 5.13. Mensaje de Error en la interfaz de acceso .....	177
Figura 5.14. Menú Sistema .....	178
Figura 5.15. Menú Módulos .....	178
Figura 5.16. Menú Catálogos .....	179
Figura 5.17. Menú Ayuda .....	180
Figura 5.18. Inicio de Interfaces para módulos/catálogos .....	183
Figura 5.19. Interfaz de Consulta .....	184
Figura 5.20. Interfaz de modificación .....	185
Figura 5.21. Eliminación de registros .....	186
Figura 5.22. Interfaz de búsqueda .....	187
Figura 5.23. Interfaz de Modificación .....	188
Figura 5.24. Vista previa para enviar a impresora .....	189
Figura 5.25. Definición del filtro .....	189
Figura 5.26. Resultado de aplicar un filtro sobre alguna columna .....	190
Figura 5.27. Habilitando la funcionalidad agrupar .....	190
Figura 5.28. Agrupación de Columnas.....	191
Figura 5.29. Agregando/Eliminando columnas en Consulta .....	192



## Bibliografía

---

Roger S. Pressman, Ingeniería del Software: Un enfoque práctico, 5ta ed;  
Ciudad de México: McGraw-Hill, 2001

Francisco Charde Ojeda, Bases de datos con Visual Basic . NET  
Madrid España: Ediciones ANAYA Multimedia 2002

Date, C. J., Introducción a los Sistemas de Bases de Datos, México: Addison Wesley  
Iberoamericana, 1986.

Ruble, David A., Análisis y Diseño Práctico de Sistemas, México: McGrawHill, 1990

Curtis L. Smith y Michael C. Amundsen, Programación de Base de Datos con Visual  
Basic, México, Prentice Hall, 1999

Burch, John G. Grudnitski, Garay, Diseño de Sistemas de Información, Teoría y  
Práctica, México: Grupo Noriega Editores.

H. M. Deitel y P. J. Deitel., Como Programar en Java, Prentice Hall, España, 1998

Luis Miguel Blanco, Programación en Visual Basic .NET. Grupo EIDOS, España, 2003

Microsoft Corporation, Manual de Referencia de Visual Basic .NET, Microsoft Press, E.U., 2002

Chris Payne, Aprendiendo ASP .NET en 21 Lecciones avanzadas, PEARSON EDUCACIÓN, México, 2002