



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**“SISTEMA DE CONTROL Y SEGUIMIENTO DE  
EXPEDIENTES EN UNA INSTITUCIÓN JURÍDICA”**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE INGENIERO EN  
COMPUTACIÓN PRESENTAN:**

**EDUARDO HERNÁNDEZ GARCÍA**

**LEONARDO HERNÁNDEZ MENDOZA**

**DIRECTOR DE TESIS:  
M.I. JUAN CARLOS ROA BEIZA**

**CIUDAD UNIVERSITARIA, MÉXICO D.F., JULIO 2013**

## **AGRADECIMIENTOS**

Un sin número de personas me han acompañado en este ciclo que hoy termina, que directa o indirectamente me han ayudado a concluirlo. No fue un camino fácil, la vida del estudiante conlleva muchos sacrificios, noches de desvelo, horas de estudio, tareas que parecen interminables, problemas con los compañeros; y si a esto le agregamos los problemas familiares que toda familia tiene, todo esto junto puede orillarnos a pensar en abandonar la escuela.

Es por ello que mis agradecimientos van, por un lado, para mi familia. Gracias a su apoyo y cariño me pude levantar de cada caída para seguir adelante, ellos que son la base de lo que soy ahora, los valores y principios que me inculcaron fueron las armas que me hacían sentir seguro de poder vencer cualquier obstáculo. Gracias padres y hermanos, sin ustedes no estaría aquí.

Por otro lado están las personas que se volvieron como de la familia, mis amigos. Esas personas extrañas en un principio que, día a día, compartiendo clases y colaborando en proyectos nos hicimos grandes amigos. Sin ustedes tampoco lo hubiera logrado, con su apoyo y compañía la escuela se convirtió en una segunda casa.

Y por supuesto que también le agradezco a los profesores y a mi amada Universidad Nacional Autónoma de México, mi orgullo y alma mater.

**LEONARDO HERNÁNDEZ MENDOZA**

Es difícil expresar el agradecimiento que tengo con todas aquellas personas que han estado presentes en mi vida y sin las cuales no sería la persona que soy ahora ni tampoco estaría en el sitio en que me encuentro en estos momentos.

En primer lugar, mi familia. Ese grupo de personas sin las que mi vida no tendría sentido y las cuales ocupan un lugar especial en mi corazón. Mi madre y mi abuela, que no sólo me han visto crecer sino que me han hecho crecer y de quienes he aprendido todo lo bueno que tengo y a superar los retos que la vida te pone. Mi hermano, a quien amo y tendré siempre como compañero de vida y amigo. A mi abuelo, primos, tíos, sobrinos, ahijados, koky y koty, quienes siempre han estado cuando los necesito y quienes me han apoyado en todo momento. Sé que la familia no se elige, pero estoy seguro que no podría haber elegido una mejor.

A mis amigos, algunos ya hermanos. Ustedes han hecho de mi vida una gran aventura y los momentos malos, una divertida historia para contar. Gracias por permitirme ser parte de sus vidas y darme tantas alegrías.

A todas aquellas personas que ya no están conmigo pero que sé que donde quiera que estén siempre me cuidan. Siempre las llevo en el corazón y en mi pensamiento. A mi amada Universidad. Le debo tanto que sé que es una deuda impagable. Amigos, educación, amor, risas, llantos, alegrías, tristezas... TODO.

A todas aquellas personas que directa o indirectamente, en mayor o menor medida han estado conmigo a lo largo de este camino.... GRACIAS.

**EDUARDO ADEMIR HERNÁNDEZ GARCÍA**

## INDICE TEMÁTICO

### CAPITULO 1 “ENTORNO DEL PROBLEMA”

1.1 Introducción.....	1
1.2 Situación actual de la institución.....	4
1.3 Conceptos básicos jurídicos.....	9

### CAPITULO 2 “MARCO TEÓRICO”

2.1 Características, ventajas y desventajas del modelo de base de datos relacional .....	10
2.2 Características, ventajas y desventajas del lenguaje de programación C# .Net. ....	20
2.3 Características, ventajas y desventajas del motor de base de datos relacionales SQL Server.....	24
2.4 Características, ventajas y desventajas de la programación en N capas. ...	29

### CAPITULO 3 “ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA”

3.1 Análisis del problema.....	39
3.2 Recopilación y análisis de la información.....	45
3.3 Requerimientos generales y particulares de la aplicación .....	51
3.4 Planteamiento de la solución y posibles módulos.....	54

<b>3.5 Justificación de la metodología y software a utilizar .....</b>	<b>60</b>
--	-----------

## **CAPITULO 4 “DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN”**

<b>4.1 El modelado del sistema .....</b>	<b>72</b>
--	-----------

<b>4.2 Implementación de Base de Datos .....</b>	<b>95</b>
--	-----------

<b>4.3 El diseño de la interfaz del usuario.....</b>	<b>112</b>
--	------------

<b>4.4 Generación de pruebas y mantenimiento .....</b>	<b>128</b>
--	------------

<b>4.5 Generación de reportes .....</b>	<b>147</b>
---	------------

<b>CONCLUSIONES.....</b>	<b>155</b>
--------------------------	------------

<b>BIBLIOGRAFÍA.....</b>	<b>157</b>
--------------------------	------------

## **CAPÍTULO I “ENTORNO DEL PROBLEMA”**

### **1.1 Introducción**

Para el beneficio de las libertades individuales de los mexicanos, el ejercicio del poder público se encuentra dividido en tres: legislativo, ejecutivo y judicial. Ver figura 1.1.1

#### **Poder Ejecutivo**

Está depositado en una sola persona, el presidente de la república, quien es electo en forma popular, directa, secreta y mayoritaria. Dura 6 años en su cargo y jamás puede volver a desempeñarlo.

Su función principal es ejecutar las leyes aprobadas por los órganos legislativos, es decir, está facultado para organizar la administración pública de acuerdo con las leyes constitucionales.

#### **Poder Legislativo**

Tiene la facultad de elaborar las leyes que rigen la vida social o de modificar las ya existentes de acuerdo con la opinión de los ciudadanos. En México, el poder Legislativo se deposita en un Congreso General (Congreso de la Unión), constituido por la Cámara de Senadores (representantes directos de los ciudadanos de los estados) y por la Cámara de Diputados.

#### **Poder Judicial**

El Poder Judicial de la Federación (P.J.F.), conforme al artículo 1° de su Ley Orgánica, se integra y ejerce por:

- La Suprema Corte de Justicia de la Nación
- El Tribunal Electoral
- Los Tribunales Colegiados de Circuito

- Los Tribunales Unitarios de Circuito
- Los Juzgados de Distrito
- El Consejo de la Judicatura Federal

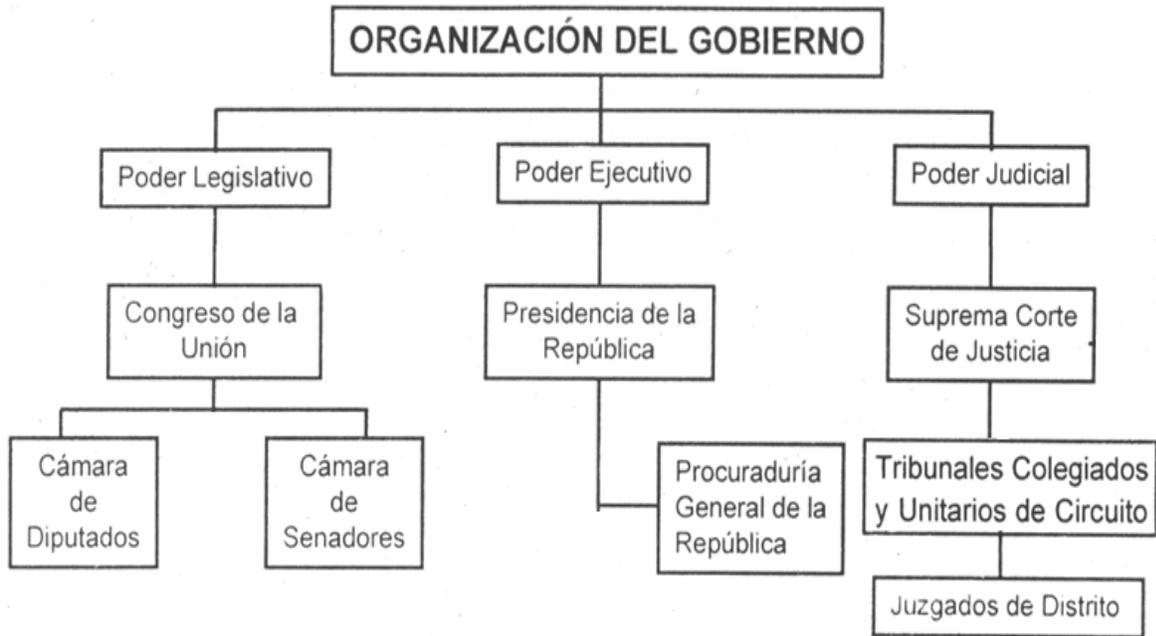
Su papel primordial lo constituye el ser interprete final de los principios y valores contenidos en la carta federal y, en este sentido, controlar la regularidad constitucional de los actos y disposiciones de las autoridades. Por estas razones, el Poder Judicial Federal representa el guardián indiscutible de la Constitución, el protector de los derechos fundamentales y el árbitro que dirime las controversias tanto entre particulares como entre poderes, generando entre ellos un equilibrio que es necesario para el sano desarrollo de la vida nacional.

Los tribunales de la federación están facultados para resolver los problemas que se produzcan por los actos u omisiones de:

- Los funcionarios que violen las garantías individuales.
- Las autoridades federales que limiten la soberanía de los estados.
- La autoridades de los estados que invadan la autoridad federal.

El Poder Judicial ha ganado espacios nunca antes ocupados en los asuntos nacionales, penetrando un conjunto de factores sociales, económicos y culturales convirtiéndolo no sólo en un ente que emite resoluciones sino en un verdadero poder político, con la creciente complejidad que conlleva.

En suma, el reto mayor al que se enfrenta el Poder Judicial consiste en dar cabal cumplimiento a la garantía constitucional que prevé la impartición de justicia en México de una manera pronta, completa e imparcial.



**Figura 1.1.1 Diagrama de organización del gobierno**

## **1.2 Situación actual de la institución**

Debido al pronunciado desarrollo demográfico, económico y tecnológico, obliga a las instituciones jurídicas a estar capacitadas para recibir, tramitar, analizar y difundir todo tipo de información jurídica para su correcto funcionamiento.

Para la institución jurídica de nuestro estudio, debido a que no cuentan con un sistema para el control y seguimiento de expedientes, el tiempo del proceso desde que un expediente se da de alta hasta que recibe una resolución final puede tardar varios días, incluso meses.

La comunicación entre las distintas áreas se da mediante correo electrónico y llamadas telefónicas, por lo que el control y búsqueda de un expediente se vuelve tedioso y tardado. El registro de la salida o entrada de un expediente en una determinada área se lleva a cabo mediante bitácoras, lo cual incrementa las probabilidades de errores humanos.

Para crear reportes estadísticos es necesaria la recopilación de información de dichas bitácoras, involucrando a todas las áreas de la institución y por ende afectando su desempeño laboral, retrasando aún más la resolución de un expediente.

Al realizar un análisis y dependiendo de la naturaleza del expediente, su origen, su jurisdicción, su materia, el tipo de recurso y otras características que la ciencia jurídica les atribuye, es posible establecer una secuencia de pasos que debe seguir desde que se ingresa hasta que se resuelve.

Dentro de los beneficios que traerá el sistema se encuentra la facilidad para conocer las diferentes fases del proceso en cualquier momento, permitiendo conocer el estado del juicio, así como el lugar donde se encuentra el expediente.

Al automatizar ciertas actividades que a los usuarios les parece fastidiosas y desgastantes se incrementará su desempeño y a la larga se les disminuirá la carga de trabajo, pudiendo atender más expedientes en menor tiempo.

Será una herramienta de apoyo a la hora de tomar decisiones, debido a que tendrán la posibilidad de consultar expedientes pasados que ya fueron resueltos y que presenten cierta similitud con nuevos casos.

Un beneficio muy importante del sistema será el poder realizar reportes estadísticos en cualquier momento que se requiera, sin quitarles tiempo laboral a los usuarios y con una mayor confiabilidad en los datos. Además de poder medir el desempeño de cada área o usuario según se requería, pudiendo identificar cuellos de botella dentro del flujo de los expedientes.

Uno de los principales beneficios además de la agilización en la tramitación de asuntos jurídico-administrativos es la disminución de la inercia burocrática y posible corrupción, lo cual alcanza niveles más significativos en el caso de la administración de la justicia, permitiendo la impartición de una justicia rápida, expedita y particularizada.

El plantear un sistema informático orientado al ámbito jurídico implica lo que todo sistema, tener un alcance bien definido, realizar un adecuado análisis de las necesidades de la institución así como de los usuarios y marcar las expectativas desde el punto de vista técnico que se tenga para con el sistema. Sin embargo al ser un sistema que servirá para una actividad específica deberá considerar aspectos relacionados con:

- Informática jurídica documental
- Informática jurídica de control y gestión
- Informática jurídica metadocumental o decisional

### **Informática jurídica documental**

Este aspecto de la informática considera la recuperación y consulta de la información pertinente a textos jurídicos, desarrollando un adecuado banco de datos que pueda ser consultado con términos propios de esta área del saber. Se han incorporado técnicas de inteligencia artificial para crear modelos que representen una interfaz entre el texto que el usuario ingresa al sistema y la información almacenada en los bancos de datos, haciendo uso de símbolos, imágenes, ideas y conceptos. Esta rama de la informática hace uso de técnicas lingüísticas, diccionarios, gramáticas generativas, etc.

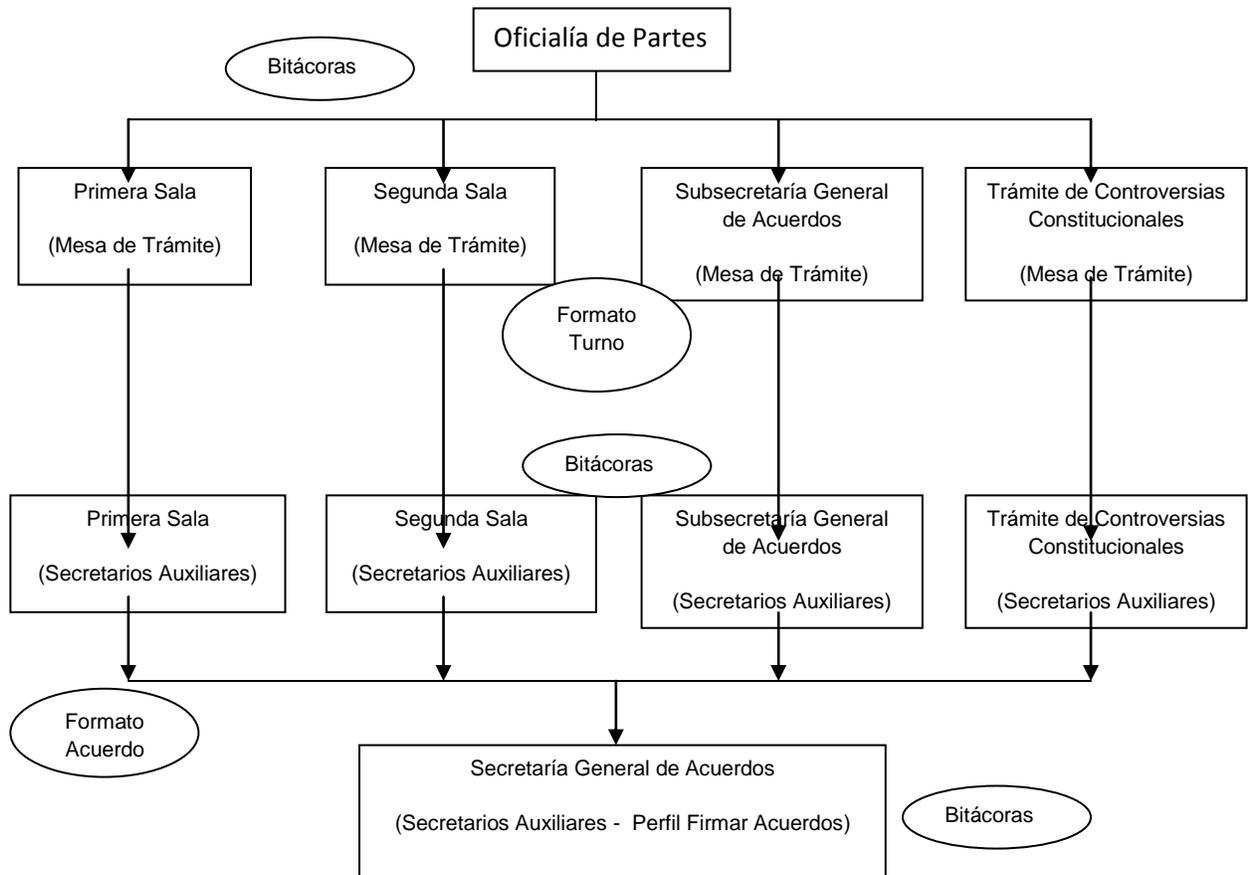
### **Informática jurídica de control y gestión**

Llamada también “ofimática”, es el resultado de la automatización de las oficinas de control y gestión que se profundizó durante toda la segunda mitad del siglo veinte con el objeto de conservar, consultar y aplicar acciones a los datos y textos, facilitando el trabajo de entidades administrativas y jurídicas, dando eficiencia a sus respectivos flujos de trabajo. Los procesadores de palabras son una herramienta que surgió del desarrollo de esta área del conocimiento.

### **Informática jurídica metadocumental o decisional**

Esta rama se concentra en aprovechar los avances de la inteligencia artificial al derecho, de tal manera que los sistemas informáticos puedan brindar soluciones a ciertas problemáticas jurídicas o apoyar en la toma de decisiones en base a mecanismos de argumentación jurídica. La solución a los casos fáciles del derecho está determinada por las normas, motivo por el cual el juzgador realiza una simple deducción silogística para llegar a ella.

## Diagrama Organizacional



**Figura 1.2.1 Diagrama organizacional**

El ciclo de vida de un expediente comienza en el área de Oficialía de Partes, se define quien lo promueve y dependiendo de su naturaleza jurídica se envía al área correspondiente. Para ésta área se creará el módulo de Ingreso de Expedientes, cuya principal función será la de crear, modificar y eliminar expedientes de manera sencilla y rápida. Además de poder consultar los registros en base a rango de fechas o demás parámetros que el usuario desee. Ver figura 1.2.1

Una vez creado el expediente y decidido a qué área corresponde su estudio y análisis, físicamente se traslada a la sub área Mesa de Trámite donde se decide a que secretario auxiliar será turnado. Para este efecto se creará el módulo

Seguimiento a Promociones, el cual filtrará únicamente los secretarios auxiliares del área correspondiente evitando así errores en el flujo que debe seguir. Contará con filtros predeterminados en los cuales se podrá consultar rápidamente qué expedientes faltan por turnar, cuales ya fueron turnados y a quién, mediante rangos de fechas. Debido a que el usuario puede cometer errores al seleccionar el secretario auxiliar, el sistema permitirá realizar un re-turno y seleccionar el correcto.

Ya que se definió que secretario auxiliar se encargará del expediente, se le entrega físicamente para que realice los acuerdos que considere necesarios y genere el documento electrónico del acuerdo si es el caso. Para ésta área se creará el módulo de Registro de Acuerdos donde podrá consultar los expedientes que le fueron turnados desde el módulo de Seguimiento a Promociones, su función principal será la de registrar estos acuerdos de manera clara y sencilla, permitiendo adjuntar el documento electrónico. Además contará con las funcionalidades de Recepción, donde podrá marcar qué expedientes físicamente le fueron entregados registrando la fecha y hora; y Devolución, para los expedientes que regresa al área anterior para una nueva valoración.

Después de crear el acuerdo, el expediente pasa al área Secretaría General de Acuerdos donde un secretario auxiliar lo estudia y si está conforme con el análisis que se le dio firma el acuerdo, terminando así el ciclo de vida del expediente. Para este propósito, dentro del módulo Seguimiento a Promociones, se agregará la funcionalidad Firmar Acuerdo y Devolver Acuerdo, ésta última servirá para cuando se regresa el acuerdo al secretario auxiliar que lo elaboró para un nuevo análisis.

El motivo de estas dos funcionalidades del módulo Seguimiento a Promociones es debido a que en éste se muestra el flujo que ha seguido el expediente, desde antes de turnarse, cuando se turna y cuando ya tiene un acuerdo, por lo que podrá tener un mejor panorama y su análisis será con datos confiables.

### 1.3 Conceptos básicos jurídicos

**Acuerdo.** En Derecho, es una decisión tomada en común por dos o más personas, por una junta, asamblea o tribunal. Es, por lo tanto, la manifestación de una convergencia de voluntades con la finalidad de producir efectos jurídicos. El principal efecto jurídico del acuerdo es su obligatoriedad para las partes que lo otorgan naciendo para las mismas obligaciones y derechos.

**Firma.** Es el nombre y apellido, o título, que una persona escribe de su propia mano. Su fin es identificar y asegurar o autenticar la identidad de un autor o remitente, o como una prueba del consentimiento y/o de verificación de la integridad y aprobación de la información contenida en un documento o similar.

**Promovente.** Viene del verbo promover y en un juzgado se refiere a la persona o institución que pide a través de un escrito el estudio de su caso, al que llamaremos promoción.

## **CAPÍTULO II “MARCO TEÓRICO”**

### **2.1 Características, ventajas y desventajas del modelo de base de datos relacional**

#### **Introducción**

Una base de datos es una colección de datos relacionados fiables y homogéneos que se encuentra agrupada o estructurada de acuerdo a un modelo de datos previamente establecido.

En esta colección, los datos deben estar estructurados de forma que reflejen fielmente los objetos, las relaciones y las restricciones existentes en el mundo real representada por la base de datos (propiedades estáticas). Para que esta representación sea fiable, la base de datos debe reflejar los cambios que los sucesos puedan provocar en el mundo real (propiedades dinámicas).

Para que un sistema pueda utilizar esta información, los datos deben ser accesibles y manipulables. Los datos deben estar organizados y almacenados en una máquina que permita el acceso en tiempo real y que asegure la confiabilidad de ellos. Estos datos deben cumplir con las siguientes propiedades:

- Están estructurados independientemente de las aplicaciones y del soporte de almacenamiento que los contiene
- Presentan la menor redundancia posible
- Son compartidos por varios usuarios y/o aplicaciones

Edgar Frank Codd en 1970 propuso el modelo relacional, el cual lo denomina como una “gestión de datos basadas en la lógica de predicados y la teoría de conjuntos utilizado para modelar problemas y administrar dinámicamente datos de forma estructurada”, por sus características, este modelo es elegido para la construcción de la mayoría de los Sistemas de Gestión de Bases de Datos comerciales.

Una base de datos relacional es un conjunto de esquemas de relación con diferentes nombres cada una y donde sus ocurrencias son las tuplas / grupos de estas relaciones.

En un modelo relacional se emplean tablas para poder organizar los elementos de datos. Cada fila o tupla representa una instancia de esa entidad y cada tabla corresponde a una entidad de aplicación. El modelo entidad relación tiene como objetivo identificar y representar de forma conceptual la información de importancia para el funcionamiento de la empresa (entidades), sus propiedades (atributos) y la forma en la que esta información se relaciona entre sí (relaciones).

En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información.

Algunos conceptos del modelo relacional son útiles para identificar a esta, a continuación se listan algunos de ellos:

- **Tupla.** Cada uno de los elementos de la relación, donde el orden de los atributos no es significativo
- **Relación.** Es una asociación significativa entre 2 o más entidades. En tal subconjunto no existirán elementos repetidos y el orden de los elementos no es significativo.
- **Dominio.** Conjunto de valores identificados por un nombre
- **Atributo.** Rol que juega un dominio dentro de una tabla, define una cualidad o propiedad de una relación.
- **Cardinalidad.** Es el conjunto de ocurrencias de una entidad en una relación.

### **Características del modelo de base de datos relacional**

Para lograr un modelo relacional en una base de datos es importante que en la fase de diseño se tomen en cuenta las características que definen a un modelo de datos relacional. Las características más importantes que se deben tomar en cuenta son:

- Independencia lógica y física de los datos
- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir 2 tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos, es un identificador único.
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y foráneas (ajenas).
- Las claves foráneas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre y por medio de éstas se hacen las relaciones.
- La información es atómica, es decir, cada campo de un registro tiene uno y sólo un valor.
- Debe haber consistencia en la información, es decir, no debe haber incongruencias en los datos que se almacenan. Frecuentemente los problemas de consistencia de datos se deben a la redundancia de éstos.
- La información debe ser NO redundante, es decir, se debe de eliminar al máximo la existencia de duplicidad de los datos.
- Los datos deben tener integridad. Se refiere a que los datos no sólo sean consistentes dentro de la base, sino además que los valores que posean los datos, sean válidos de acuerdo a las dependencias funcionales entre tablas y de acuerdo a las políticas de negocio. La integridad de la base se puede lograr mediante:
  - El mantenimiento de una redundancia mínima y controlada.
  - El establecimiento de llaves primarias.
  - La validación de dependencias entre las tablas relacionadas.
  - La creación de reglas de validación durante la inserción y edición de datos.
- Se debe asegurar que sólo los usuarios que estén autorizados para acceder a los datos puedan hacerlo y que lo puedan hacer en cualquier momento, es decir, que sea accesible la información en cualquier momento.

## **Ventajas del modelo de base de datos relacional**

Por otro lado, una base de datos relacional debe cumplir con las siguientes condiciones para brindar la fiabilidad y la seguridad al momento de crear un sistema:

- Tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento.
- Los datos son compartidos por diferentes usuarios y aplicaciones.
- Tanto datos como procedimientos pueden ser transportables conceptualmente a través de diferentes Sistemas de Gestión de Bases de Datos.

Un Data Base Management System, en español Sistema de Gestión de Base de Datos (SGBD), es una herramienta de software que permite la creación y la manipulación de una base de datos.

Proporciona el método de organización necesaria para el almacenamiento y recuperación de grandes cantidades de datos. Todo sistema de gestión de base de datos está basado en un modelo de datos. Estos proporcionan estructuras de datos predefinidas con sus operadores asociados (modelos clásicos), o bien mecanismos de estructuración de datos o constructores de tipos más generales.

Otras ventajas del modelo son las siguientes:

- Simplicidad y generalidad al momento de la planeación y la esquematización de la base de datos.
- Facilidad de uso para el usuario final.
- Control sobre la redundancia de datos.
- Consistencia de datos.
- Compartición de datos.
- Facilidad para poder hacer consultas de información.
- Los estándares que rigen el modelo de base de datos relacional están creciendo, son conocidos y utilizados por los fabricantes de SGBD's y por los desarrolladores.
- Integridad de datos mediante las reglas o restricciones que se establezcan.

- Flexibilidad y soporte para tipos de datos complejos.
- Favorece la normalización por ser más comprensible y aplicable.
- Favorece la accesibilidad a los datos debido a la estructuración de sus entidades y sus relaciones.
- Las búsquedas de información se hacen más sencillas y rápidas.
- Se incrementa la productividad al poder insertar, eliminar y consultar registros de una manera rápida y con pocas instrucciones.
- Se aumenta la concurrencia al poder dar acceso a varios usuarios al mismo tiempo.
- La curva de aprendizaje para poder crear esquemas entidad-relación son relativamente cortos.

### **Desventajas de las bases de datos relacionales**

Los datos provenientes de archivos individuales se integran en un repositorio centralizado, como puede ser una base de datos, donde se pierde el sentido de propiedad. Es decir, perdemos también la responsabilidad por los datos. Como consecuencia los datos inexactos pueden pasar inadvertidos.

Esto puede causar serios problemas a menos que se tomen medidas adecuadas para la integridad y veracidad de datos. La base de datos también se puede convertir en blanco de fallas de seguridad a menos que se mantenga una disciplina estricta. Más aún, una base de datos puede acrecentar conflictos políticos organizativos dentro de una empresa, debido a que sirve a las necesidades de múltiples usuarios, algunos de los cuales pueden tener conflictos legítimos de intereses y diferentes necesidades.

La seguridad de la información también puede llegar a ser un problema, ya que al ser un sistema concurrente, la información se compromete y de no tener reglas estrictas y específicas para todas las posibles eventualidades se puede llegar a tener problemas de seguridad.

## Álgebra Relacional

El álgebra relacional es un conjunto de operaciones simples sobre tablas relacionales, a partir de las cuales se definen operaciones más complejas.

Dichas operaciones utilizan una o dos relaciones existentes para crear una nueva relación. Esta nueva relación puede entonces usarse como entrada para una nueva operación. El álgebra relacional se apoya en dos tipos de operaciones, las unarias (sólo interviene un elemento) y las binarias (intervienen dos elementos). Las operaciones del álgebra relacional son las siguientes:

- **Selección.** También conocida como consulta, consiste en recuperar un conjunto de registros de una tabla o de una relación indicando las condiciones que deben cumplir los registros recuperados. En dichas consultas se utilizan diferentes operadores de comparación ( $=$ ,  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $<>$ ), además de los operadores lógicos (and, or, xor, not).
- **Unión.** Es utilizada para poder recuperar datos a través de varias tablas conectadas unas con otras mediante cláusulas de unión, en cualquiera de sus tres variantes, unión por la derecha, unión por la izquierda y unión completa (derecha e izquierda).
- **Proyección.** Es un caso específico de la selección. Es una segunda selección, en la que se seleccionan solo aquellos campos que se desean obtener.
- **Asignación.** Consiste en asignar un valor a uno o varios campos de una tabla.
- **Diferencia.** Consiste en recuperar los registros que cumplan con pertenecer a una relación y que no pertenezcan a otra relación del mismo esquema.

## Cálculo Relacional

El cálculo relacional es un lenguaje de consulta que describe la respuesta deseada sobre una Base de Datos sin especificar cómo obtenerla, a diferencia del álgebra

relacional que recurre a diferentes tablas y realiza varios pasos para obtener un resultado, el cálculo realiza la operación en un único paso gracias a lenguajes de interrogación de Bases de Datos; aunque ambos métodos siempre pueden llegar a lograr los mismos resultados.

La solución para toda consulta en este tipo de cálculo se define por:

- Una lista de resultados: Son aquellos registros que cumplen las condiciones que desean.
- Una sentencia de cualificación: Contiene las condiciones que deseamos que cumplan los registros de la lista de resultados.

El cálculo relacional incluye un concepto denominado cuantificador, los cuantificadores averiguan el número de registros afectados por una determinada operación, incluso antes de realizarla. Los cuantificadores se pueden dividir en:

- Existenciales: Buscan el número de registros que devolverá una consulta.
- Universales: Indican que una condición se aplica a todas las filas de un tipo.

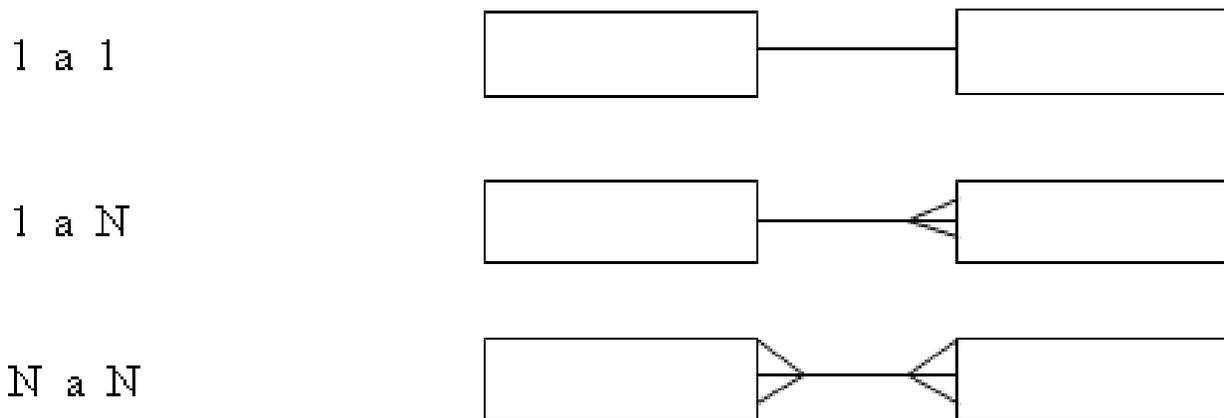
El cálculo relacional se apoya en las siguientes operaciones para obtener una respuesta sobre la Base de Datos:

- Reunión: Permite unir datos de varias relaciones.
- Diferencia: Identifica filas que están en una relación y no en otra.
- Intersección: Permite unir campos idénticos pertenecientes a las filas que son comunes en dos relaciones.
- Producto: Consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles combinaciones entre los registros de la primera y los registros de la segunda.
- Las combinaciones de formulas se generan a partir del uso de operadores como NOT, AND, OR.

## Cardinalidad

La cardinalidad es simplemente la forma en la que se relacionan las Entidades, expresa cuántas cantidades se relacionan con otras entidades. Las estructuras de datos según la cardinalidad son las siguientes:

- **Uno a uno (1:1).** Cada instancia o elemento de la unidad A está asociado solamente a un elemento de la unidad B.
- **Uno a muchos (1:M).** Cada instancia o elemento de la unidad A está asociado a varios elementos de la unidad B, entonces la clave que forma el vínculo entre ambas entidades, pasa hacia la entidad que tiene mayor grado de cardinalidad, es decir, el que posee la denominación “muchos”,
- **Muchos a muchos (N:M).** Los elementos de la entidad A están asociados a varios elementos de la entidad B, y los elementos de la entidad B están asociados a varios elementos de la entidad A, cuando esto sucede se genera una nueva entidad denominada “Entidad Asociada”. Ver figura 2.1.



**Figura 2.1. Diagrama de estructura de datos**

## **Normalización**

La teoría de la normalización está basada en la observación de que en cierto conjunto de relaciones son más eficaces las operaciones de inserción, actualización y eliminación de datos que las que tendrían otros conjuntos de relaciones conteniendo los mismos. Es un proceso que pretende conseguir tablas con una estructura óptima y eficaz.

En dicho proceso se somete un esquema de relación a una serie de pruebas para “certificar” si pertenece o no a una cierta forma normal. Esta se basa en las dependencias funcionales entre los atributos de una relación.

La normalización de los datos puede considerarse como un proceso de análisis de los esquemas de relación de datos basado en sus claves primarias para alcanzar las propiedades deseables de minimizar la redundancia y minimizar las anomalías de inserción, eliminación y actualización de datos. Los esquemas de relación insatisfactorios que no cumplan determinadas condiciones como las pruebas de formas normales, se descomponen en esquemas de relación más pequeños que satisfagan dichas pruebas y que de este modo posean las propiedades deseables.

### **Primera Forma Normal 1FN**

Es la más elemental de todas, se definió para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones. Una relación está en primera forma normal si todo atributo contiene un valor atómico para cada uno de sus elementos, es decir, que una estructura en forma de tabla solo puede tener un valor en cada intersección de renglón y columna. Ningún grupo de valores repetido está permitido en la intersección. La eliminación de ese grupo de datos implica crear más tablas.

## **Segunda Forma Normal 2FN**

Una relación está en segunda forma normal si y solo si: la relación está en primera forma normal y si todo atributo que no pertenezca a la llave primaria es completamente dependiente de manera funcional de la llave completa. Es decir, todo atributo que no sea llave necesita en forma completa la llave para ser identificado de manera única. La segunda forma normal permite eliminar las redundancias para que ningún atributo este determinado por una parte de una llave. Toda estructura de tipo tabla en la segunda forma normal implica necesariamente que se encuentra en la primera forma normal.

## **Tercera Forma Normal 3FN**

Una relación está en tercera forma normal si y solo si: la relación está en segunda forma normal y si además todo atributo que no pertenece a la llave no depende de otro atributo que no es llave. Cuando un atributo que no es llave se puede determinar con uno o más atributos que son llave, se dice que existe dependencia transitiva entre ambos, esto significa que una estructura de tipo tabla está en tercera forma normal si no existe dependencia transitiva entre los atributos que no son llaves. La tercera forma normal permite asegurar la eliminación de redundancia debida a las dependencias transitivas.

## **Cuarta Forma Normal 4FN**

La 4NF se asegura de que las dependencias multivaluadas independientes estén correcta y eficientemente representadas en un diseño de base de datos. La 4NF es el siguiente nivel de normalización después de la forma normal de Boyce-Codd.

Una tabla está en 4NF si y solo si esta en Tercera forma normal o en BCNF (Cualquiera de ambas) y no posee dependencias multivaluadas no triviales. La definición de la 4NF confía en la noción de una dependencia multivaluada. Una tabla

con una dependencia multivaluada es una donde la existencia de dos o más relaciones independientes muchos a muchos causa redundancia; y es esta redundancia la que es suprimida por la cuarta forma normal.

### **Quinta Forma Normal 5FN**

También conocida como forma normal de proyección-uniión (PJ/NF), es un nivel de normalización de bases de datos designado para reducir redundancia en las bases de datos relacionales que guardan hechos multi-valores aislando semánticamente relaciones múltiples relacionadas. Una tabla se dice que está en 5NF si y sólo si está en 4NF y cada dependencia de unión (join) en ella es implicada por las claves candidatas.

## **2.2 Características, ventajas y desventajas del lenguaje de programación C#**

**.Net.**

### **Introducción**

El lenguaje de programación C# es un lenguaje orientado a objetos seguro y elegante que permite a los desarrolladores construir aplicaciones robustas y seguras. Los creadores de este lenguaje quisieron mejorar a sus antecesores, el lenguaje C y C++, incorporando también las ventajas y mejoras del lenguaje JAVA. Así se consiguió que tuviese las ventajas del C, del C++, pero además la productividad que posee el lenguaje JAVA y se le denominó C#.

El lenguaje C# fue pensado para desarrollar aplicaciones empresariales aunque también se pueden desarrollar aplicaciones del tipo: aplicaciones de cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, etc.

Aunque el lenguaje fue creado específicamente para usarse mediante el Framework .NET, no solo se pueden crear aplicaciones Windows con este lenguaje de

programación. Existen diversas Interfaces que nos permiten trabajar con C# para crear aplicaciones para diferentes tipos de sistemas operativos, incluso aquellos diseñados para dispositivos móviles.



Figura 2.2.1 Logotipo C#

### Características del lenguaje de programación C#

- Sintaxis expresiva, sencilla y fácil de aprender.
- Se encuentra bajo los estándares de **ECMA** e **ISO**.
- Está basado en signos de llave como **C++**, **Java**, **AS3**, **PHP** o **JavaScript**.
- Tiempo de aprendizaje DEMASIADO breve para los que estén familiarizados con **C++**, **Java**, **AS3**, **PHP** o **JavaScript**.
- Simplifica muchas de las complejidades de **C++**.
- Proporciona características eficaces como tipos de valor que admiten valores *NULL*, delegados, expresiones *lambda*, enumeraciones y acceso directo a memoria. Características que no se encuentran en **Java** de manera total.
- Admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos.
- Admite iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente.

- Las expresiones *Language-Integrated Query* (**LINQ**) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.
- **NETAMENTE orientado a objetos**, cumpliendo con todas sus características tales como:
  - Encapsulación.
  - Herencia.
  - Polimorfismo.
  - Todo atributo y método (incluidos los constructores y Main o .STARTUP) se encapsulan dentro de la definición de una clase.
  - Una clase puede heredar de una única clase y puede implementar cualquier número de interfaces.
  - Las redefiniciones o definiciones de métodos que son virtuales en la clase padre, requieren la palabra clave *override* como medio para evitar redefiniciones accidentales.
  - Admite estructuras o *struct* que son como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.
- Además de los principios básicos de orientación a objetos, facilita el desarrollo de Software implementando varias construcciones innovadoras de lenguaje como las siguientes:
  - Firmas de métodos encapsulados denominadas *delegados*, que habilitan notificaciones de eventos con seguridad de tipos.
  - Ya no hay necesidad de *métodos de accesibilidad*, **C#** nos trae *propiedades*, que actúan como descriptores de acceso para variables miembro privadas.
  - Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
  - Comentarios en línea de documentación XML.
  - *Language-Integrated Query* (**LINQ**) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

- **INTEROPERABILIDAD**

- Puede interactuar directamente con software de Windows como objetos **COM** o archivos **DLL** nativos de **Win32**.
- Habilita los programas de C# para que puedan realizar prácticamente las mismas tareas que una aplicación C++ nativa.
- Admite incluso el uso de punteros y el concepto de código “no seguro” en los casos en que el acceso directo a la memoria sea totalmente crítico.
- El proceso de compilación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, structs, interfaces y eventos.

### **Ventajas del lenguaje de programación C#**

Además de las características que se acaban de mencionar en el punto anterior, la mayor ventaja que posee el lenguaje es que está específicamente diseñado a la medida para el framework .NET y que la manera en que se programa la interfaz Visual Studio es muy intuitiva, fácil y con todas las facilidades para el programador.

El soporte, las comunidades, manuales y ejemplos que se pueden encontrar son bastante grandes y mejoran en mucho la experiencia de programación. Hay una comunidad de programadores C# muy grande y sigue creciendo.

También el lenguaje en sí ya es utilizado como framework para crear sistemas en otras plataformas. Un ejemplo de esto, es la plataforma XNA que se utiliza como herramienta para la creación de videojuegos en la consola X-BOX, en el cual el lenguaje nativo de la plataforma es C#.

## **Desventajas del lenguaje de programación C#**

Las desventajas que se derivan del uso de este lenguaje de programación son que en primer lugar se tiene que conseguir una versión reciente de Visual Studio .NET, la cual en la mayoría de las versiones no es gratuita. Por otra parte se deben cumplir con algunos requerimientos mínimos del sistema para poder trabajar adecuadamente.

Además para quien no está familiarizado con ningún lenguaje de programación, le costará más trabajo iniciarse en su uso, y si se quiere consultar algún tutorial más explícito sobre la programación en C# se tendría que contar además con una conexión a Internet.

## **2.3 Características, ventajas y desventajas del motor de base de datos relacionales SQL Server.**

### **Introducción**

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional (SGBDR o RDBMS, Relational Database Management System) diseñado para trabajar con grandes cantidades de información y tiene la capacidad de cumplir con los requerimientos de proceso de información para aplicaciones comerciales y sitios web.

Es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, éstas a su vez están compuestas por filas y columnas. En el centro de SQL Server está el motor SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.

SQL Server incluye herramientas para la administración de los recursos que el ordenador nos proporciona y los gestiona para un mejor rendimiento de la base de datos.

Una buena instalación y configuración de SQL Server, y sobre todo una buena administración de las herramientas que éste nos proporciona, logrará:

- Qué las consultas que se realicen mediante sentencias SQL obtengan un tiempo de respuesta óptimo.
- Qué la memoria y la CPU de la máquina estén aprovechadas al máximo.

Transact-SQL es el lenguaje que utiliza SQL Server para poder enviar peticiones tanto de consultas, inserciones, modificaciones, y de borrado a las tablas, así como otras peticiones que el usuario necesite sobre los datos. En definitiva, es un lenguaje que utiliza SQL Server para poder gestionar los datos que contienen las tablas.

El lenguaje estándar SQL (Structured Query Language) se emplea para los sistemas de bases de datos relacionales RDBMS (Relational Database Management System), es el estándar ANSI (American National Standards Institute). También es utilizado por otros sistemas como: Oracle, Access, Sybase, etc.

A lo largo de los años éste sistema gestor ha tenido varias versiones, siempre tratando de innovar y mejorar en el ámbito de las bases de datos. Vea tabla 2.1.

<b>Versión</b>	<b>Año</b>	<b>Nombre de la versión</b>	<b>Nombre clave</b>
1.0 (OS/2)	1989	SQL Server 1-0	SQL
4.21 (Win NT)	1993	SQL Server 4.21	SEQUEL
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	HYDRA
7.0	1998	SQL Server 7.0	SPHINX
-	1999	SQL Server 7.0 OLAP Tools	PLATO
8.0	2000	SQL Server 2000	SHILOH

8.0	2003	SQL Server 2000 64-bit Edition	LIBERTY
9.0	2005	SQL Server 2005	YUKON
10.0	2008	SQL Server 2008	KATMAI
10.50	2010	SQL Server 2008 R2	KILIMANJARO
11	2012	SQL Server 2012	DENALI

**Tabla 2.1 Versiones SQL Server**

Para éste proyecto utilizaremos la versión SQL Server 2008 R2 por ser la versión estable más reciente y con las características más avanzadas y probadas en este momento.



**Figura 2.3.1 Logotipo SQL Server 2008 R2**

### **Características SQL SERVER 2008 R2**

- Protección de la base de datos mediante métodos de encriptación avanzada para archivos de datos o archivos log sin necesidad de realizar cambios en las aplicaciones.
- Administración extensible de clave que ofrece una solución completa para administrar encriptaciones y claves, incluso para claves creados por terceros.
- Reduce la complejidad del espejado de bases de datos permitiendo que las organizaciones recuperen el failover principal por medio de la desactivación temporaria del espejado dentro del mecanismo de failover.

- Compresión de archivos log mejorada para ofrecer un rendimiento óptimo y reducir el ancho de banda utilizado en la red.
- Respuesta consistente y predecible con el Administrador de Recursos. Esta función permite que las organizaciones definan límites y prioridades de recursos para diferentes cargas de trabajo, lo cual ayuda a que las cargas de trabajo actuales les ofrezcan a los usuarios finales un rendimiento consistente.
- Los datos se almacenan de una manera más eficiente, y reduce los requisitos de almacenamiento para los datos. La Compresión de Datos también ofrece mejoras significativas en el rendimiento para grandes cargas de trabajo I/O.
- Permite administrar las políticas de una o varias instancias de SQL Server 2008.
- Incluye una colección de datos de rendimiento más extensa, un nuevo repositorio de datos centralizados para almacenar datos con máximo rendimiento, y nuevas herramientas para reportes y monitoreos.
- Ayuda a que los desarrolladores traten las solicitudes de datos con un lenguaje de programación administrado, por ejemplo C# o VB.NET, en lugar de las declaraciones de SQL. Admita solicitudes orientadas a configuración, escritas en lenguajes .NET, para que se ejecuten contra ADO.Net (LINQ a SQL), ADO.Net DataSets (LINQ a DataSets), el Marco de Entidades de ADO.NET (LINQ a Entidades) y el proveedor de Entity Data Service Mapping. Utilice el nuevo proveedor LINQ a SQL que permite que los desarrolladores utilicen LINQ directamente sobre tablas y columnas de SQL Server 2008.
- Hace que las aplicaciones de bases de datos modelen estructuras en árbol de una manera más eficiente. Un nuevo tipo de sistemas –una ID con jerarquía– puede almacenar valores que representan nodos en una jerarquía en árbol.
- Permite que grandes datos binarios se almacenen directamente en un sistema de archivos NTFS mientras se preserva una parte integral de la base de datos y mantiene una consistencia transaccional. Activa la escalada de grandes datos binarios tradicionalmente administrados por la base de datos para almacenarlos fuera de la base de datos, en un sistema de almacenamiento más económico.

- Los datos NULL no consumen espacio físico, lo cual supone una manera muy eficiente de administrar datos vacíos en una base de datos.
- Elimina el límite de 8KB para UDTs (User Defined Types o Tipos Definidos de Usuarios), lo cual permite que los usuarios amplíen el tamaño de su UDTs.
- Se necesita menos capacidad de almacenamiento para tener backups online y para que estos backups se ejecuten significativamente rápido ya que se necesita menos disco I/O.
- Con SQL Server 2008 realiza análisis más completos con capacidades analíticas mejoradas y con computaciones y agregados más complejos. Las nuevas herramientas de diseño ayudan a los usuarios a ajustar el desarrollo de la infraestructura de análisis, permitiéndoles generar soluciones para un rendimiento optimizado.

## **Ventajas de SQL SERVER 2008 R2**

- Madurez de un sistema que lleva muchos años innovando y desarrollando dentro del área de los sistemas gestores de bases de datos.
- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.
- Interactúa con las herramientas de desarrollo Microsoft de una manera fácil, amigable y transparente.
- Está soportado por una de las empresas más poderosas y con mayor alcance dentro del ámbito de las TI.

## **Desventajas de SQL SERVER 2008 R2**

- Es una herramienta NO gratuita.
- Los requerimientos del sistema para su instalación son bastante altos.
- MSSQL usa Address Windowing Extensión (AWE) para hacer el direccionamiento de 64-bit. Esto le impide usar la administración dinámica de memoria, y sólo le permite alojar un máximo de 64 GB de memoria compartida.
- MSSQL no maneja compresión de datos (excepto la versión 2008 Enterprise Edition, que sí lo hace), por lo que las bases de datos pueden llegar a ocupar mucho espacio en disco.
- MSSQL requiere de un sistema operativo Microsoft Windows, por lo que no puede instalarse, por ejemplo, en servidores Linux, por esta razón.

## **2.4 Características, ventajas y desventajas de la programación en N capas.**

### **Introducción**

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño. Para poder entender este concepto es necesario explicar la arquitectura cliente-servidor que es la base fundamental para la programación por capas.

### **Arquitectura Cliente-Servidor**

La arquitectura Cliente-Servidor también conocida como modelo Servidor-Cliente es una forma de dividir y especializar programas y equipos de cómputo, a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia y permita simplificarlas. Con ésta tecnología, estamos frente a la plataforma abierta por excelencia ya que es el procesamiento cooperativo de la información por medio de

un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

En esta arquitectura la capacidad de proceso esta repartida entre el servidor y los clientes. Los clientes interactúan con el usuario, usualmente en forma grafica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Una de las clasificaciones mejor conocidas de las arquitecturas Cliente/Servidor se basa en la idea de planos (tier), la cual es una variación sobre la división o clasificación por tamaño de componentes. Esto se debe a que se trata de definir el modo en que las prestaciones funcionales de la aplicación serán asignadas, y en qué proporción, tanto al cliente como al servidor. Dichas prestaciones se deben agrupar entre los tres componentes clásicos para Cliente/Servidor: interfaz de usuario, lógica de negocios y los datos compartidos, cada uno de los cuales corresponde a un plano.

Dentro de esta categoría tenemos las aplicaciones en dos planos (dos capas), tres planos (tres capas) y multi-planos (n-capas). Este término ha sido utilizado indistintamente para referirse tanto a la parte lógica (software) como a la parte física (hardware) de una aplicación.

La parte a la que se refiere la programación en n-capas se da lugar en la parte lógica de la aplicación, este segmento es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura Cliente/Servidor y en la madurez y popularidad de la computación distribuida. Por ejemplo, esto permite hablar de servidores de aplicación distribuidos a lo largo de una red, y no tiene mucho sentido identificar a un equipo de hardware como servidor, sino más bien entenderlo como una plataforma física sobre la cual pueden operar uno o más servidores de aplicaciones.

Los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas cliente/servidor:

- El proceso cliente es quien inicia el dialogo. También es conocido como Front-End.
- El proceso servidor es el que pasivamente espera a que lleguen peticiones de servicio y se le conoce con el término de Back-End
- El Middleware corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

### **Características del modelo cliente-servidor**

- El Cliente y el Servidor pueden actuar como una sola entidad y tambien pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas o en la misma plataforma.
- Un servidor da servicio a múltiples clientes.
- Cada plataforma puede ser escalable independientemente.
- Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por remplazo tecnológico, se realizan de una manera transparente para el usuario final.

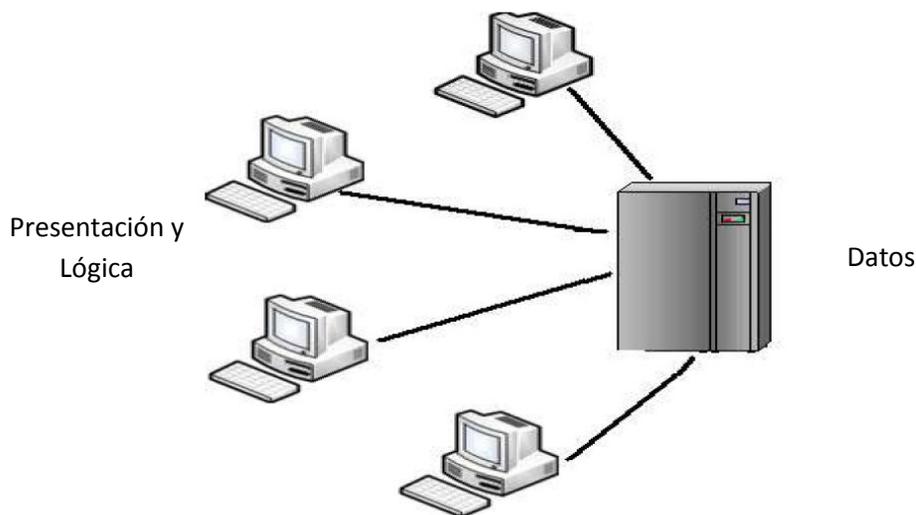
### **Modelo Cliente - Servidor de Dos Capas**

Tradicionalmente la arquitectura cliente / servidor está basada en un modelo de computación de dos capas. Este modelo consiste de un cliente y un servidor de base

de datos. El procesamiento de tareas y la lógica de la aplicación son compartidas entre el servidor de base de datos y el cliente.

En este modelo a los clientes se les llama clientes pesados, en donde reside mucho del poder de procesamiento y de la lógica de la aplicación. Esto hace que el mantenimiento del cliente sea costoso.

Adicionalmente, los clientes pueden estar operando en diferentes plataformas, dando como resultado una distribución compleja de versiones de aplicaciones específicas de las plataformas.



**Figura 2.4.1 Modelo Cliente-Servidor de 2 capas.**

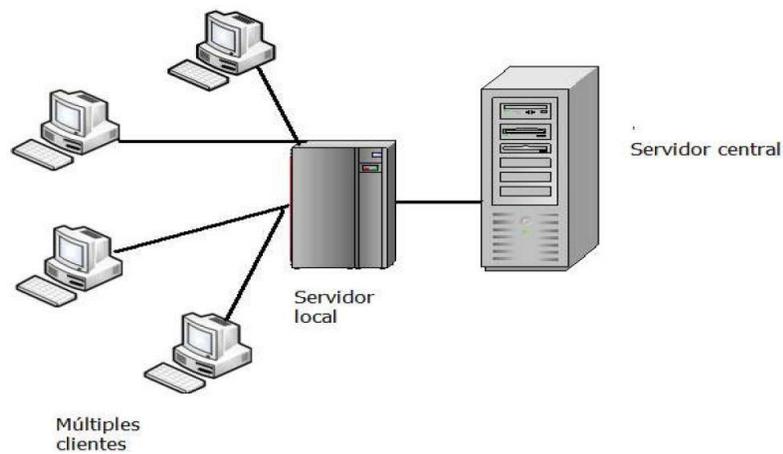
Las desventajas del modelo de dos capas son:

- El nivel de las aplicaciones se recargan, entremezclando aspectos típicos del manejo de la interfaz con las reglas del negocio.
- Las reglas del negocio quedan dispersas entre el nivel de aplicación y los procedimientos de la base de datos. El nivel de aplicación puede ser demasiado pesado para el cliente.
- También puede existir en este modelo un cliente delgado y un servidor robusto; esto es lo que generalmente se utiliza en la práctica.

## **Modelo Cliente - Servidor de Tres Capas.**

El modelo de computación de tres capas es un modelo evolucionado del modelo de dos capas. En este modelo, existe una capa intermedia entre el cliente y el servidor de base de datos. Esta capa consiste de un servidor de aplicaciones que contiene el grueso de la lógica de la aplicación. Los clientes en este modelo son clientes livianos o clientes ligeros. Con esta arquitectura la lógica de la aplicación reside en una sola capa que puede ser fácilmente mantenida. El diseño arquitectónico de la capa media puede también ser optimizada en funciones del servidor puesto que éste no tiene que contener u hospedar la base de datos.

En esta arquitectura de tres capas, el software del cliente (capa cliente) es ligeramente pesado, suficiente para ser descargado bajo demanda y lo suficientemente pequeño como para presentar la interfaz del usuario. El grueso de la lógica de la aplicación está implementada ya sea en la capa media (servidor de aplicaciones) o está almacenada en la base de datos. Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación. Algo muy importante y que es la mayor ventaja de esta arquitectura es que ahora únicamente se cambia la regla en el servidor de aplicación y ésta actuará en todos los clientes, cosa que no sucedía con la arquitectura en dos capas ya que si alguna regla se cambia, es necesario ir a cada cliente a realizar el cambio. La mayor desventaja que tiene esta arquitectura es que no se puede realizar un repositorio de datos distribuido, un ejemplo de este tipo de repositorio es el de los bancos que necesitan de un repositorio de datos distribuido para poder tener sus sucursales en otros países y ciudades.



**Figura 2.4.2 Modelo Cliente Servidor de tres capas**

### **Modelo Cliente-Servidor de n-capas**

El modelo n-tier (n-capas) de informática distribuida ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma en la mayor parte de las empresas. Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes.

Todas las aplicaciones basadas en n-capas permitirán trabajar con clientes ligeros, tal como navegadores de Internet, WebTV, Teléfonos Inteligentes, PDAs (Personal Digital Assistants o Asistentes Personales Digitales) y muchos otros dispositivos preparados para conectarse a Internet.

De este modo, las arquitecturas de n-capas se están posicionando rápidamente como la piedra angular de los desarrollos de aplicaciones empresariales y las compañías están adoptando esta estrategia a una velocidad de vértigo como mecanismo de posicionamiento en la economía emergente que tiene su base en la red (lo que se ha venido a denominar "Nueva Economía").

Las arquitecturas basadas en n-capas permiten a los componentes de negocio correr en una LAN, WAN o Internet. Esto significa que cualquiera con un ordenador y conexión a la Red (por ejemplo, un miembro del equipo de fuerza de ventas) posee toda la funcionalidad que tendría si se encontrase delante de su sistema de escritorio.

Realmente, los componentes distribuidos de una arquitectura de n-capas es una tecnología esencial para crear la siguiente generación de aplicaciones e-business, aplicaciones que son altamente escalables, fiables y que proporcionan un alto rendimiento y una integración sin fisuras con los sistemas de back-end heredados.

Las arquitecturas empresariales de n-capas se están convirtiendo en la nueva base para el desarrollo de aplicaciones de misión crítica y ofrecen la única arquitectura funcional para la siguiente generación de soluciones informáticas distribuidas basadas en Internet. Los sistemas distribuidos de n-capas proporcionan un conjunto de avances tecnológicos sin precedentes, como pooling de conexiones, multiplexado de conexiones, balanceo de carga dinámico y rendimientos excelentes en hardware trabajando en cluster. Las aplicaciones industriales basadas en n-capas pueden ayudar a las compañías a desarrollar un nuevo núcleo de habilidades en prácticamente todo, desde la gestión del conocimiento hasta los sistemas relacionados con comercio electrónico.

A diferencia de lo que se pudiera pensar, el desarrollo en n-capas no es un producto o un estándar, es un concepto estratégico que ayuda a la construcción y despliegue lógico de un sistema distribuido. Los sistemas de n-capas subdivididos ayudan a facilitar el desarrollo rápido de aplicaciones y su posterior despliegue, con beneficios incrementales fruto de los esfuerzos del desarrollo en paralelo coordinado y del outsourcing inteligente, resultando un enorme decremento del tiempo de desarrollo y de sus costes.

Una arquitectura basada en clientes ligeros desplaza la capa de presentación de la aplicación en el lado del cliente, mientras que la lógica de negocio y los datos residen en el middleware y los servidores de back-end. El diseño para clientes ligeros minimiza los problemas de despliegue de las aplicaciones, mientras que maximiza la accesibilidad a la misma desde una amplia variedad de plataformas heterogéneas. Los frameworks basados en n-capas se crean para obtener las ventajas de los estándares abiertos de la industria que permiten a las aplicaciones resultantes operar en entornos distribuidos multiplataforma.

Utilizando estos potentes estándares abiertos se permite a los Integradores de Sistemas asegurar la efectividad y consistencia de la comunicación y mensajería a todos los niveles en aplicaciones cruzadas Interdepartamentales e Interempresariales.

El diseño de aplicaciones basado en n-capas considera a la red como un pool de servicios distribuidos, un concepto mucho más ambicioso que el simple acceso de un cliente a un servidor. La separación de la presentación, lógica de negocio y datos es realizada en un número indefinido de capas lógicas, permitiendo a cada capa ser desarrollada, mejorada, gestionada y desplegada de forma independiente. Esta es precisamente la base para el modelo de informática de red en n-capas. Las plataformas multicapa funcionan consistentemente a lo largo de un variado conjunto de hardware, permitiendo escalar las operaciones del negocio desde un simple portátil, hasta un datacenter, desde el dispositivo más simple hasta el más complejo de los mainframes.

Las arquitecturas basadas en n-capas son el siguiente paso lógico en un proceso de evolución, el cuál, está basado en las arquitecturas convencionales cliente-servidor (2 y 3 capas) más la convergencia de dos tecnologías tremendamente potentes:

- Desarrollo de aplicaciones basadas en componentes - relacionado directamente con la Programación Orientada a Objetos (Lenguajes y Técnicas)
- Internet - primer ejemplo de un sistema complejo de n-capas cliente-servidor.

Los sistemas de n-capas utilizan técnicas de desarrollo basadas en componentes combinados con los estándares abiertos de Internet, para crear aplicaciones multiplataforma muy potentes con bajos costes, fáciles de mantener y con gran efectividad. Lo que realmente es nuevo en el modelo de n-capas es la posibilidad de distribuir objetos independientes sobre el número de capas que sean necesarias y enlazarlas dinámicamente, cuando sea necesario, para proporcionar una flexibilidad ilimitada a la aplicación.

N-Tier forma parte también de un revolucionario proceso, actualmente en desarrollo, basado en la aplicación de estas nuevas tecnologías (componentes y estándares de Internet). Estas tecnologías son los bloques para crear Software de Negocio y Sistemas de Información adaptables que ayuden a las empresas a integrar todos sus sistemas de Tecnologías de la Información, así como las inversiones realizadas en éstos, mientras que obtienen una ventaja clara en el uso de Internet.

El surgimiento de la tecnología de componentes distribuidos es la clave de las arquitecturas de n-capas. Estos sistemas de computación utilizan un número variable de componentes individuales que se comunican entre ellos utilizando estándares predefinidos y frameworks de comunicación como:

- CORBA - (Common Object Request Broker Architecture) del Object Management Group (OMG)
- DNA - (Distributed interNet Architecture) de Microsoft (incluye COM/DCOM y COM+ además de MTS, MSMQ, etc.)
- EJB - (Enterprise Java Beans) de Sun Microsystems
- XML - (eXtensible Markup Language) del World Wide Web Consortium (W3C)

Estas y otras tecnologías en rápida evolución proporcionan la infraestructura necesaria y la fontanería relacionada que permite a las compañías operar en un entorno complejo, multiplataforma y con capacidades de computación distribuida, tanto interna como externamente según se requiera en cada caso.

El desarrollo de aplicaciones en n-capas es un proceso iterativo de división del problema en piezas manejables denominadas componentes. Estos componentes, o "Componentes de Negocio - Business Objects" son "modelos software" basados típicamente en la "vista" de un objeto real, evento o proceso de negocio. Los componentes software individuales pueden formar parte y adaptarse tanto de estructuras independientes como de sistemas colaborativos.

El diseño de aplicaciones en n-capas es ideal para la creación de sistemas adaptables, donde cada componente puede ser utilizado y reutilizado en nuevas combinaciones para satisfacer requisitos de negocio dinámicos. Esto permite a los desarrolladores y a las nuevas aplicaciones reutilizar componentes existentes que modelan lógica de negocio sobradamente probada. En un entorno tremendamente cambiante como el actual, utilizar aplicaciones basadas en diseños de n-capas posibilita a las empresas ser más ágiles y adaptables en proporcionar valor a sus clientes. Los sistemas basados en n-casas tienen el potencial de reducir drásticamente tanto el time-to-market para las nuevas aplicaciones de negocio, como el coste total de mantenimiento, adaptando estos complejos y caros sistemas a las siempre cambiantes necesidades empresariales.

Los framework de n-capas ayudan a proveer una solución integral enlazando los procesos de negocio críticos a la vez que se conservan las inversiones existentes de la empresa en tecnologías de la información y recursos relacionados.

## **CAPITULO 3 “ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA”**

### **3.1 Análisis del problema**

El hecho de contar con un sistema informático integral donde se llevará un registro y control de expedientes ayudará a evitar problemas como:

- Pérdida de documentos
- Duplicidad de expedientes
- Desconocimiento del flujo que ha seguido
- Errores humanos (reducirlos lo más posible)
- Excesivo tiempo en resolver un caso
- Poca comunicación entre las distintas áreas
- Falta de pruebas para deslindar responsabilidades
- Falta de elementos contundentes para poder medir el desempeño de los trabajadores

Nuestra institución de estudio se encuentra dividida en 4 diferentes áreas para poder tratar en su totalidad un caso, dicha separación de actividades ayuda a que el flujo se vuelva mecánico y en teoría más ágil.

Tenemos entonces las siguientes áreas:

#### **Oficialía de Partes**

Es donde se da inicio el ciclo de vida del expediente. La función principal de ésta área es definir, dependiendo de la naturaleza del caso, a qué área será enviado para su oportuno estudio.

#### **Mesa de Trámite**

Aquí lo que se hace es asignar el expediente a un secretario auxiliar en particular, confirmando que el expediente debe ser tratado en dicha o si debe regresarse a

Oficialía de Partes para que se envíe a donde corresponda. El expediente no debe ser asignado a un secretario auxiliar de un área distinta a la que fue enviado.

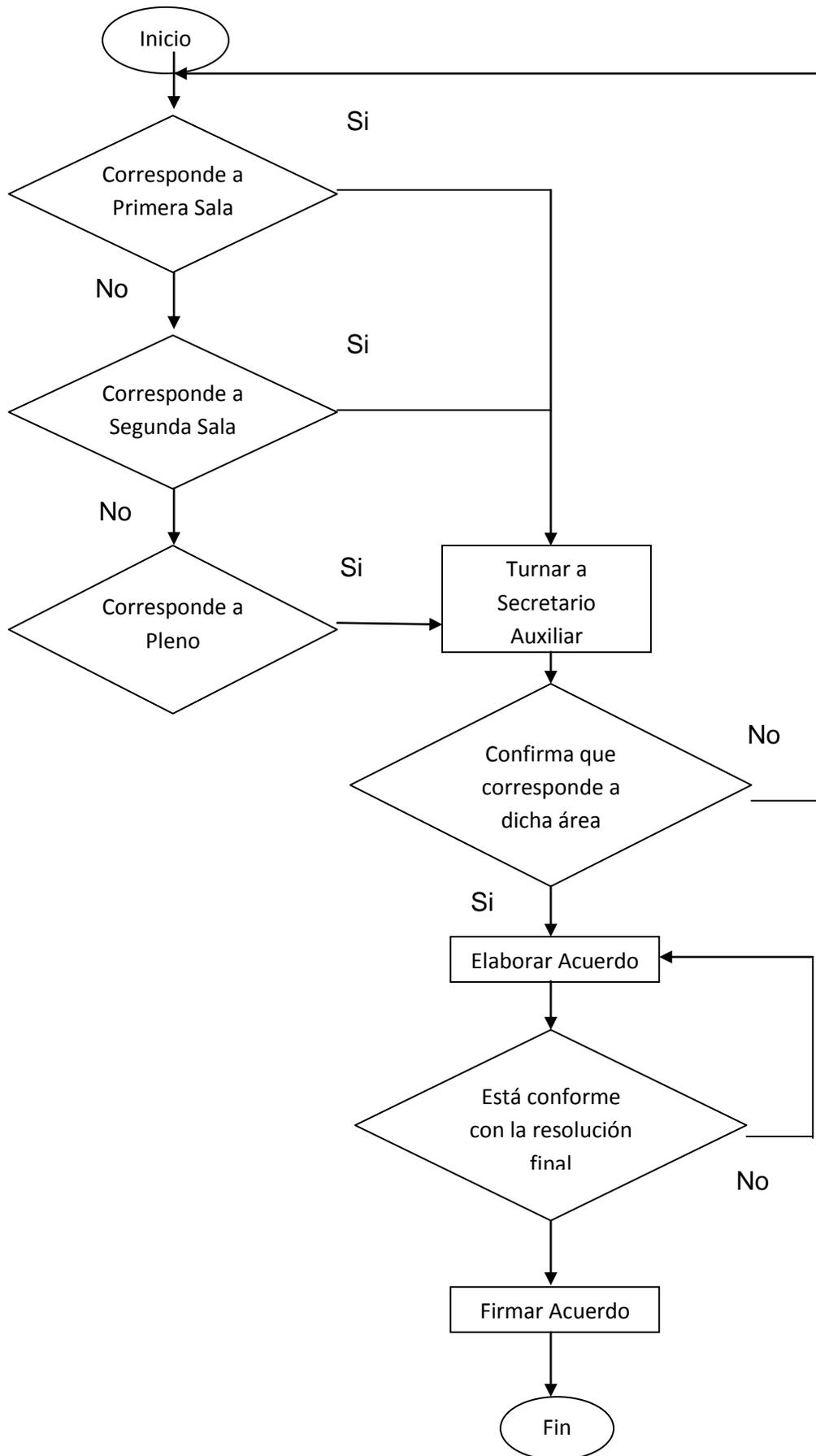
### **Registro de Acuerdos**

Una vez que el expediente fue turnado, dicho secretario auxiliar debe recibir el expediente físicamente para que después del análisis correspondiente realice el acuerdo que a su juicio sea el indicado. El secretario auxiliar puede basarse en expedientes anteriores como forma de apoyo, dado que puedan presentar algunas similitudes con el actual.

### **Secretaría General de Acuerdos**

Ya que se tiene el acuerdo, el cual puede contar con un documento electrónico, los secretarios auxiliares de ésta área deben firmar dicho acuerdo como forma de aprobación de todo el flujo del expediente y la resolución final. En caso de no estar de acuerdo con ésta parte final, lo devuelven al área anterior para un nuevo análisis.

A partir de éste análisis de nuestra institución, podemos obtener el siguiente diagrama:



Recabando información con los usuarios y analizando a detalle el flujo de los expedientes pudimos darnos cuenta de algunos problemas que repercuten en el tiempo total de su ciclo de vida.

## **Oficialía de Partes**

Dentro de ésta área que es la que da inicio al expediente encontramos los siguientes problemas:

- Incorrecta definición del área que tratará el expediente
- Duplicidad de expedientes
- Errores ortográficos
- Tiempo que toma consultar los catálogos para el llenado de ciertos campos

El hecho de definir incorrectamente el área a la que compete el análisis del expediente repercute en un tiempo muerto, ya que será enviado y después devuelto para una nueva valoración, y nuevamente enviado al área ahora si correspondiente. Se vuelve un doble trabajo y se va retrasando su resolución final.

Un problema grave en ésta área es la duplicidad de expedientes, no sólo porque puede generar una doble carga de trabajo y desperdicio de papel, sino porque se podrían dar resoluciones distintas lo cual afectaría la credibilidad de la institución.

Si hay algo que puede ayudar de manera sencilla a acelerar el tratamiento de un expediente es reducir las posibilidades que tenga un usuario de cometer errores ortográficos. En el ámbito legal tiene una gran relevancia y puede generar graves problemas una falla de este tipo.

Dentro de la generación del expediente inicial, hay ciertos campos que deben ser llenados en base a catálogos preestablecidos. El usuario los consulta desde un archivo tipo excel y va generando el documento, con la posibilidad de equivocarse en los campos y el tiempo que esto lleva.

## **Mesa de Trámite**

Aquí encontramos los siguientes contratiempos:

- Incorrecta asignación del secretario auxiliar que tratará el expediente
- Falta de certeza en la correcta asignación del área
- Desconocimiento del tiempo que ha pasado desde que fue asignado a dicha área, hasta el punto en que es recibido
- Comunicarle al secretario auxiliar que le fue turnado dicho expediente
- Desconocimiento de cuantos expedientes fueron enviados a dicha área

En base a un catálogo de secretarios auxiliares, el usuario selecciona el que se encargará del análisis y posterior acuerdo. Pero puede llegar a ocurrir que le asigne uno que se encuentra en otra área, generando un verdadero conflicto en cuanto al flujo que debía seguir. El secretario auxiliar lo regresará para su correcto turno y se perderá tiempo en este proceso.

Para saber si el expediente fue enviado al área correcta, se debe hacer un análisis de la naturaleza del expediente, lo cual lleva tiempo. Para poder realizar éste análisis de forma adecuado es necesaria una persona que tenga conocimientos sólidos sobre aspectos legales, lo cual provoca que no cualquier usuario pueda validar dicha asignación.

Cuando en un área en específico reciben un expediente, no saben en realidad cuanto tiempo ha pasado desde que fue creado y les fue asignado. Es posible que el expediente por error haya sido entregado en otra área y que cuando llegue a la correcta ya hayan pasado días. Si el usuario supiera este tiempo muerto que ha pasado, podría darle prioridad y reducir en algo el tiempo final.

El secretario auxiliar sabe que le fue turnado un expediente debido a que le llega físicamente. Debe realizar un análisis preliminar para cerciorarse que le fue enviado correctamente, lo cual lleva tiempo.

Si cada área quisiera realizar un reporte estadístico de cuantos expedientes le son enviados y cuantos turna a cada secretario, sería una labor de varios días y que afectaría el tiempo laboral de los usuarios.

## **Registro de Acuerdos**

El proceso de generar los acuerdos presenta los siguientes detalles:

- Respaldo el documento del acuerdo creado
- Poder consultar los acuerdos creados en otros expedientes, en forma de apoyo
- Falta de certeza en cuanto al número de expedientes a los que les tiene que crear el acuerdo
- Falta de comunicación con la siguiente área donde se realiza la firma

El acuerdo es un documento electrónico que el secretario auxiliar genera, exponiendo ahí los criterios y razonamientos por los cuales fue creado. El único respaldo que tiene es el que se guarda en la máquina del usuario, por lo que si el equipo llega a fallar o se borra dicho archivo por error, no quedará respaldo electrónico.

Para poder consultar los acuerdos creados con anterioridad por otros secretarios auxiliares, es necesario enviar un correo electrónico para pedir una copia de dicho documento, lo cual puede tomar tiempo en ser respondido.

Si el usuario pudiera consultar rápidamente cuantos expedientes tiene que tratar al día podría organizar mejor su tiempo, atendiendo con prioridad a los más atrasados e ir disminuyendo su carga de trabajo de manera más organizada.

El problema de comunicación entre las áreas es similar que en los procesos anteriores, el secretario que realiza la firma se entera de qué expedientes tiene que analizar hasta que los recibe físicamente.

## **Secretaría General de Acuerdos**

El proceso de firmar el acuerdo es en teoría más sencillo, sin embargo encontramos los siguientes detalles:

- Falta de certeza en cuanto al número de expedientes que se tienen que firmar.
- No hay registro de los acuerdos firmados y los devueltos para su nuevo análisis

Si el usuario de ésta área pudiera medir con certeza cuántos acuerdos firmó y cuantos le quedaron pendientes, podría tener una mejor planeación de sus actividades e incrementar su rendimiento. Por ello se vuelve importante tener un registro sobre su trabajo realizado durante el día.

### **3.2 Recopilación y análisis de la información**

Para comenzar a planear el diseño y las posibles funcionalidades que tendrá nuestro sistema es necesario conocer los formatos, catálogos, bitácoras y todo aquello que sirve de apoyo al usuario para ir construyendo los expedientes.

Gracias a esta recopilación y análisis de las distintas fuentes de información podemos comenzar a trabajar en los siguientes puntos:

- Diseño de la base de datos
- Funcionalidades generales
- Funcionalidades particulares
- Diseño de la interfaz gráfica
- Número aproximado de aplicaciones que serán necesario desarrollar

Uno de los puntos más importantes de nuestro sistema será el diseño de la base de datos. Si no tomamos en cuenta todos los posibles escenarios y reglas de negocio con que se maneja nuestra institución de estudio, nuestras tablas, campos y relaciones pueden resultar incoherentes y por ende el sistema será inoperante.

Es posible que entre las distintas áreas de la institución encontremos actividades que sean idénticas, lo cual nos ayudará a definir las funcionalidades generales que compartirán todas las aplicaciones. Esto es muy importante para la etapa de desarrollo del sistema, ya que será código que será común en los distintos módulos.

Las funcionalidades particulares vendrán del análisis de cada una de las áreas, por lo que la comunicación con sus usuarios será clave para obtener los requerimientos que más satisfagan sus necesidades. Al final de cuentas lo que se busca es ayudarles a realizar su trabajo lo más rápido posible.

Una interfaz gráfica amigable siempre aportará a favor en el desempeño de los usuarios, por lo cual el diseño será tomado en cuenta con la importancia necesaria, basada en estándares y normas, y siempre tomando en cuenta la opinión del usuario.

Una vez que tengamos una idea aproximada de las funcionalidades y diseño de la interfaz, podremos hacer el cálculo de cuantas aplicaciones serán necesarias, qué funcionalidades tendrá cada una, y la organización de las mismas.

Para poder realizar esto se recopilaron los siguientes formatos y catálogos.

Este es el formato para dar de alta un nuevo expediente, ver figura 3.2.1

Formato de Alta de un nuevo expediente				
<b>Datos Generales</b>				
Tipo de Asunto				
No. Expediente				
Destino				
Número de Registro				
Estado de la Federación				
Municipio				
Materia				
Sub Materia				
<b>Promoventes</b>				
Tipo de Promovente				
Promoventes				
<b>Autoridades</b>				
Autoridades Demandadas				
Autoridades responsables del cumplimiento de la sentencia				
<b>Tema</b>				
Tema Planteado				
Observaciones				

**Figura 3.2.1 Formato “alta de acuerdo”**

Catálogos en los que se apoyan los usuarios para elegir secretarios auxiliares y los tipos de acuerdo. Ver figuras 3.2.2 y 3.2.3

Catálogo Secretarios Auxiliares	
SecretarioAuxiliar	Genero
(1) MARIO ISMAEL MARTÍNEZ ROMERO.	H
(2) ARCELIA BUENDÍA CANTO.	M
(2) FERNANDO ORDAZ HERNÁNDEZ.	H
(9) MAURICIO GUERRERO MARTÍN.	H
(3) LUIS BOLAÑOS MARTÍN	H
(4) RAQUEL RAMÍREZ VERA.	M
(9) ALEJANDRO SÁNCHEZ HERNÁNDEZ.	H
(2) SOLEDAD NÁJERA PAREDES	M
(7) IGNACIO ROBLES SOTO	H
(10) MARÍA ESTHER ESPINOZA BUENDÍA	M
(8) ELSA GUTÉRREZ OLGÍN	M
ADRIANA CAMPUZANO GALLEGOS	M
MANUEL CARAVANTES SÁNCHEZ	H
ANDRÉS MARTÍNEZ MARTÍNEZ	H
CARLOS ENRIQUE GÓMEZ MARTÍNEZ	H
FRANCISCO JAVIER QUIÑONES PEÑA	H
GUADALUPE IVONNE BRIONES ESQUIVEL	M
LUIS GERARDO VILCHIS PÉREZ	H
MARÍA ESTHER SUÁREZ HUITRÓN	M
FRANCISCO SANDOVAL LÓPEZ	H
ROCIO BALDERAS FERNÁNDEZ	M
JORGE HUMBERTO BENÍTEZ PIMIENTA	H
LUZ CUETO MARTÍNEZ	M
JUAN CARLOS CRUZ RAZO	H
ROBERTO LARA HERÁNDEZ	H
ADRIANA ESCORZA CARRANZA	M
JORGE QUEZADA MENDOZA	H

**Figura 3.2.2 Catalogo secretarios auxiliares**

Catálogo Tipo de Acuerdo	
TipoAcuerdo	
AVOCAMIENTO	
TRAMITE	
FALLO	
TURNO A PONENTE	
AUTO INICIAL	
AUTO PRESIDENCIA	
AUTO DE IMPROCEDENCIA	
COMPETENCIA DELEGADA	
RADICACIÓN	
CON SELLO	
AGREGADO	
PREVENCIÓN	
ACLARACIÓN DE DEMANDA	
ADMISIÓN DE LA DEMANDA	
DESECHAMIENTO DE DEMANDA	
CONTESTACIÓN DE DEMANDA	
ADMISIÓN DE LA AMPLIACIÓN DE DEMANDA	
PEDIMENTO PGR	
OPINIÓN DE PGR	
CONCEDE SUSPENSIÓN	
NIEGA SUSPENSIÓN	
DESAHOGA VISTA	
CUMPLE REQUERIMIENTO	
CONTESTA AMPLIACIÓN DE DEMANDA	

**Figura 3.2.3 Tipos de acuerdo**

El catálogo de tipo de asunto es muy importante, ya que dependiendo de este campo es el área a la cual es asignado el expediente. Ver figura 3.2.4

Catálogo Tipo de Asunto		
TipoAsunto	NombreCorto	Abreviatura
AMPARO DIRECTO	Amp. Dir.	A.D.
AMPARO EN REVISIÓN	Amp. Rev.	A.R.
CONFLICTO COMPETENCIAL	Conf. Comp.	CONF.COMP.
CONTRADICCIÓN DE TESIS	Contr. Tesis	C.T.
CONTROVERSIA POR EXTRADICIÓN	Contr. Extrad.	C.EXTRADICION
RECONOCIMIENTO DE INOCENCIA	Rec. Inoc.	REC.INOCENCIA
QUEJA	Quejas	QUEJA
VARIOS	Varios	VARIOS
CONTROVERSIA CONSTITUCIONAL	Cont. Const.	C.C.
AMPARO DIRECTO EN REVISIÓN	Amp. Dir. Rev.	A.D.R.
CONSULTA	Consulta	CONSULTA
DENUNCIA DE REPETICIÓN DEL ACTO F	Den. Repet.	D.R.A.R.
EXPEDIENTES EN EJECUCIÓN	Exp. en Ejecución	EXPEDIENTES EN EJ.
IMPEDIMENTO	Impedimentos	IMPEDIMENTO
INCIDENTE DE INEJECUCIÓN DE SENTEN	Inc. Inejecución	I.I.S.
INCONFORMIDAD	Inconformidad	INCONFORMIDAD
RECURSO DE RECLAMACIÓN	Reclamaciones	R.R.
QUEJA ADMINISTRATIVA	Queja Admva.	Q.A.
ACCIÓN DE INCONSTITUCIONALIDAD	Acc. Inconst.	A.I.
REVISIÓN ADMINISTRATIVA	Rev. Admva.	R.A.
DICTAMEN FINAL DE SOLICITUD DE EJE	Dictamen Final Art. 27 Const.	ART.97
JUICIO DE INCONFORMIDAD	Juicio inconf.	J.I.
SOLICITUD DE EJERCICIO DE LA FACULT	Art. 97	ART.97
FACULTAD DE ATRACCIÓN	F. de Atracción	S.E.F.A.
ACLARACIÓN DE SENTENCIA	Aclaración de sentencia	ACLARA.SENT.
CUMPLIMIENTO DE CONVENIOS DE CO	Cumpto. de Conv. de Coord. Fiscal	C.C.F.
ARTÍCULO 11 DE LEY ORGÁNICA PJF FR	Art.11-IX Ley Orgánica PJF	ART.11.PJF

**Figura 3.2.4 Catálogo de tipos de asunto**

El catálogo de Órganos es sumamente extenso y resulta tardado encontrar el que se requiere. Ver figura 3.2.5

Catálogo Órganos		
Organo	Domicilio	Residencia
NO DEFINIDO	NULL	NULL
SEGUNDA SALA	PINO SUAREZ	DISTRITO FEDERAL
PRIMERA SALA	PINO SUAREZ	DISTRITO FEDERAL
PLENO	PINO SUAREZ	DISTRITO FEDERAL
TRIBUNAL COLEGIADO EN MATERIAS PENAL Y DEL TRABAJO DEL OCTAVO CIRCUITO		
TRIBUNAL COLEGIADO DEL DÉCIMO CIRCUITO	AV. UNIVERSIDAD KM 12, RESERVA TERRIT	COATZACOALCOS
NOVENO TRIBUNAL COLEGIADO DE CIRCUITO	???	CUERNAVACA
TERCER TRIBUNAL COLEGIADO DE CIRCUITO DE	???	SALTILLO
SEGUNDO TRIBUNAL COLEGIADO DE CIRCUITO	EDIFICIO SEDE DEL PJF CANCÚN, AV. ANDR	CANCÚN
PRIMER TRIBUNAL COLEGIADO DE CIRCUITO DE	EDIFICIO SEDE DEL PJF CANCÚN, AV. ANDR	CANCÚN
SEXTO TRIBUNAL COLEGIADO DE CIRCUITO DE	BLVD. DEL LAGO NO. 103, VILLAS DEPORTIV	CUERNAVACA
QUINTO TRIBUNAL COLEGIADO DE CIRCUITO DE	BLVD. DEL LAGO NO. 103, VILLAS DEPORTIV	CUERNAVACA
CUARTO TRIBUNAL COLEGIADO DEL DÉCIMO C	BLVD. DEL LAGO NO. 103, VILLAS DEPORTIV	CUERNAVACA
TERCER TRIBUNAL COLEGIADO EN MATERIAS C	CARRETERA CUATRO CARRILES GUANAJUA	GUANAJUATO
SEXTO TRIBUNAL COLEGIADO DE CIRCUITO DE	PALACIO FEDERAL AV. MADERO ORIENTE N	MORELIA
QUINTO TRIBUNAL COLEGIADO DE CIRCUITO DE	EDIFICIO SEDE DEL PODER JUDICIAL AV. CA	MORELIA
CUARTO TRIBUNAL COLEGIADO EN MATERIAS	CULTURAS VERACRUZANAS NO 120, LA RES	XALAPA
TRIBUNAL COLEGIADO SEGUNDO CIRCUITO	AVENIDA BORDO DE XOCHIACA NO. 2 "C",	NEZAHUALCÓYOTL
CUARTO TRIBUNAL COLEGIADO DE CIRCUITO DE	ADMINISTRACIÓN DE LOS EDIFICIOS SEDE	DF
TERCER TRIBUNAL COLEGIADO DE CIRCUITO DE	ADMINISTRACIÓN DE LOS EDIFICIOS SEDE	DF
TRIBUNAL COLEGIADO EN MATERIAS PENAL Y	BOULEVARD ISIDRO LÓPEZ ZERTUCHE NO.	SALTILLO
TRIBUNAL COLEGIADO EN MATERIAS PENAL Y	CENTRO DE JUSTICIA FEDERAL. AV. MALECON LEANDRO	
TRIBUNAL COLEGIADO EN MATERIAS PENAL Y	AV. HERÓICO COLEGIO MILITAR NO. 204, R	OAXACA
TRIBUNAL COLEGIADO EN MATERIAS PENAL Y	EDIFICIO SEDE DEL PODER JUDICIAL DE LA F	MÉRIDA
TRIBUNAL COLEGIADO EN MATERIAS DE TRAB	CALLE VIOLETAS NO. 1007, REFORMA, OAX	OAXACA
TRIBUNAL COLEGIADO EN MATERIAS DE TRAB	EDIFICIO SEDE DEL PODER JUDICIAL DE LA F	MÉRIDA
TRIBUNAL COLEGIADO EN MATERIAS CIVIL Y T	BOULEVARD INDEPENDENCIA 2111 ORIENT	TORREÓN
TRIBUNAL COLEGIADO EN MATERIAS CIVIL Y D	CENTRO DE JUSTICIA FEDERAL. AV. MALECÓN LEANDRO	

**Figura 3.2.5 Catálogo de órganos**

Al observar ésta información podemos empezar a planear los tipos de controles que utilizaremos, su organización en las pantallas, el orden más amigable para el usuario según como ha venido trabajando hasta ahora, así como los posibles íconos a diseñar. Desde este punto ya se podrían empezar a crear las tablas que serán sólo de consulta y servirán para llenar los controles tipo “combobox”.

### **3.3 Requerimientos generales y particulares de la aplicación**

Un alto porcentaje de los programas de computadora modernos a los que se enfrenta el usuario se caracterizan por un aspecto en común: ponen a su disposición una o varias ventanas que le permiten recibir o mostrar los datos que se procesan en el sistema. Esta característica es la que ha permitido que más personas vean la computadora como una herramienta de trabajo casi indispensable en más y más tareas que deben realizar en su vida cotidiana.

Pero, lo que realmente ha facilitado el uso del computador no han sido las ventanas como tal, sino el diseño de las mismas incluyendo unos elementos básicos que hoy en día se han convertido en prácticamente un estándar del desarrollo de las interfaces gráficas de usuario. Los botones de comando, las cajas de texto, las listas desplegables, las barras de herramientas, las barras de desplazamiento, los menús, y muchos otros elementos son parte indispensable en cualquier programa de computador que el usuario busca le sea útil para sus objetivos.

Hoy en día al usuario de aplicaciones de software no le interesa en que sistema operativo funcionen estas, y menos en que lenguaje de programación hayan sido desarrolladas, siempre y cuando pongan a su disposición los elementos básicos que él reconoce, de tal manera que le permitan acceder a las funciones en la misma forma como lo hacen desde las aplicaciones más elementales hasta las más sofisticadas.

Uno de los aspectos que puede resultar más complejo para un programador es la ubicación de los componentes en varias ventanas, que le obligan a memorizar diferentes coordenadas o realizar cálculos que en la mayoría de los casos, aunque importantes en el campo de la presentación, resultan intrascendentes para la funcionalidad misma del programa que se está desarrollando.

Por todo esto es primordial la comunicación con los que serán los usuarios del sistema, ya que de ello dependerá la precisión de los requerimientos y el diseño. Podemos dividirlos según sus características en generales y particulares:

### Requerimientos Generales

- Ambiente Microsoft Windows XP o superior
- Aplicación con ventana principal
- Ventanas llamativas
- Botones con iconos
- Ingresar a la aplicación mediante un usuario y contraseña
- Acceso directo desde el escritorio
- Validar conexión de red

Todos los usuarios cuentan con equipos con sistema operativo Microsoft Windows desde la versión XP hasta Windows 7 y se encuentran muy familiarizados con sus tipos de aplicaciones, es por ello que el sistema debe ser desarrollado con ventanas, botones y demás controles característicos de este sistema operativo.

Las aplicaciones que se ejecutan en Windows generalmente están constituidas por una o varias ventanas, pero siempre va existir una ventana que actúa como la base de las demás, o lo que podemos llamar una ventana principal. Esta ventana principal es quien se encarga de servir como base para que el usuario pueda controlar toda la aplicación.

Esa parece ser la lógica del cerebro humano, las cosas son más fáciles si existe un centro de mandos desde donde se pueda interactuar con todas las funcionalidades que tiene un sistema.

Aunque pueden programarse aplicaciones donde no haya ninguna ventana que actúe como centro de control, se ha comprobado que generalmente este tipo de aplicaciones resulta de difícil comprensión y manejo hasta para los usuarios más experimentados.

Para el usuario es muy importante el aspecto visual, y un buen uso de los colores nos sería de gran ayuda. Aunque a simple vista pudiera pensarse una tarea sencilla, es necesario encontrar un balance entre colores llamativos y aquellos que puedan lastimar la vista del usuario.

Los botones muy probablemente serán los controles que más utilizará el usuario, es por ello que mientras más rápido identifique la funcionalidad de cada uno, la manipulación del sistema será la ideal. Con la creación de íconos representativos de su función buscaremos cumplir con este objetivo.

Sólo el usuario autorizado deberá poder usar el sistema por lo que es necesario contar con una ventana inicial donde deberá ingresar un usuario y contraseña que lo autentique.

#### Requerimientos particulares

- Filtrar por fecha
- Búsqueda de palabras
- Exportar registros a archivos tipo Word y Excel
- Ordenar por columna
- Cambiar tipo de letra y tamaño de los registros
- Teclas de acceso rápido para los botones

El hecho de que el usuario pueda filtrar los registros por fecha le permitirá llevar un mejor control de su trabajo, dándole prioridad a los expedientes que lleven más tiempo retrasados.

La búsqueda de palabras será una de las funcionalidades de mayor apoyo, ya que permitirá fácilmente encontrar los expedientes que se requieran. Con estas dos herramientas pretendemos que el usuario tenga una mayor manipulación de los registros.

Para efectos de generar reportes impresos se creará la funcionalidad de exportar a archivos de tipo Word y Excel. Será en un formato predefinido por el usuario y con la posibilidad de mostrar los campos que el usuario elija.

El sistema permitirá ordenar los registros según la columna que elija el usuario, con esto podrá organizar a su conveniencia la información y generar reportes mejor estructurados.

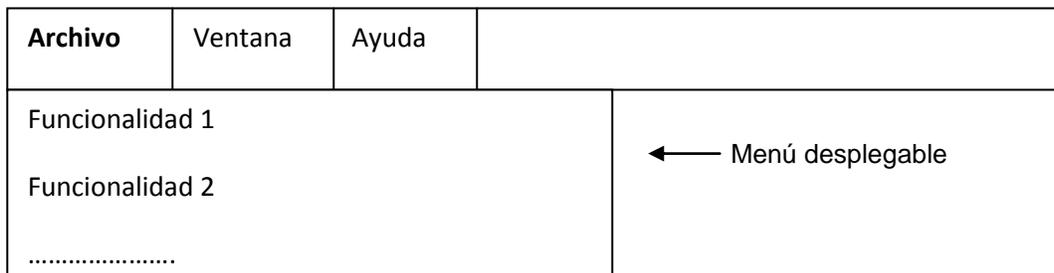
Debido a los diferentes gustos de los usuarios en cuanto al tamaño y tipo de letra, el sistema permitirá elegir el de su preferencia y quedará guardado para cada uno, así cuando vuelva abrir la aplicación no tendrá que volver hacerlo.

Una característica muy común en las aplicaciones tipo Windows son las teclas de acceso rápido para los botones, las cuales suelen ayudar a automatizar el uso de las distintas funcionalidades.

### 3.4 Planteamiento de la solución y posibles módulos

Tomando en cuenta los requerimientos generales y particulares de la aplicación, y en base a nuestra propia experiencia en el uso de distintas herramientas podemos crear un bosquejo sobre el posible diseño y distribución de los componentes.

Menú general

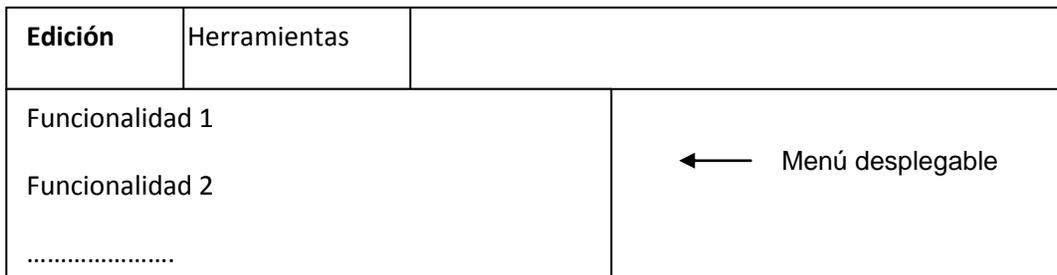


**Figura 3.4.1 Menú general**

El menú general (Ver figura 3.4.1) sería una barra de herramientas en la parte superior de nuestra ventana principal, éste menú contendría funcionalidades que serían comunes entre las distintas aplicaciones. Los botones que contendría esta barra serían Archivo, Ventana y Ayuda.

En el botón Archivo tendríamos funcionalidades como salir del sistema, abrir y cerrar ventanas, etc. En el de Ventana serían las que nos ayudarían a acomodar las ventanas según sea necesite, ya sea vertical, horizontal o en cascada. Para el de Ayuda tendríamos ahí alguna guía sobre cómo usar la aplicación.

**Menú Principal de la aplicación**



**Figura 3.4.2 Menú principal de la aplicación**

Este menú se enlazaría con el menú general y estarían en la misma barra de herramientas, contendría funcionalidades propias de cada aplicación con los posibles botones de Edición y Herramientas. Dentro de Edición estarían las acciones más importantes y de mayor uso para el usuario, mientras que en Herramientas serían funcionalidades de apoyo. Ver figura 3.4.2

**Menú de acceso rápido**



**Figura 3.4.3 Menú de acceso rápido**

Este menú sería una barra de botones justo debajo del menú principal, contendría todas las funcionalidades del sistema acomodándolas según sean particulares o generales. Sería la forma más rápida de acceder a ellas para el usuario. Cada uno de los botones tendría un ícono representativo de su función, para mayor facilidad de distinción. Ver figura 3.4.3

#### Menú de búsqueda

<input type="text"/>	Anterior	
----------------------	----------	--

**Figura 3.4.4 Menú de búsqueda**

Estaría compuesto por un cuadro de texto para introducir lo que se desea buscar, con los botones de Anterior y Siguiente para irse moviendo entre las coincidencias que encuentre dicha búsqueda. Su ubicación sería en la parte superior, a la altura del menú principal, alineada a la derecha. Ver figura 3.4.4

#### Menú de Filtros

Filtro Predefinido	Fecha Desde	Fecha Hasta
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Figura 3.4.5 Menú de filtros**

En éste menú tendremos un combo box con los filtros predefinidos, esto es, condiciones de búsqueda que arrojarán los registro de mayor importancia para el usuario.

Para acotar los resultados según se requiera, tendremos dos controles que servirán para que el usuario escoja las fechas que delimiten el rango de búsqueda. Ver figura 3.4.5

### Área para visualizar datos

Título referente a los datos					
Columna 1	Columna 2	.....	.....	.....	Columna n

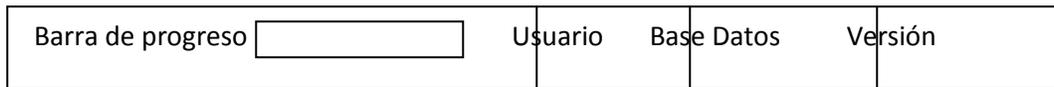
Fila seleccionada / Total de filas

**Figura 3.4.6 Área para visualizar datos**

Aquí es donde se mostrarán los datos que resulten de la búsqueda y del filtro seleccionado. La idea es que se muestren en celdas que permitan una clara visualización de su contenido. Para esto, dependiendo del tipo de dato será el ancho de la celda y el usuario podrá modificar el tamaño a lo alto.

Para facilidad de selección de los renglones tendremos una columna para este propósito, ocupando un control de tipo check box. Es muy importante que al usuario le quede claro qué registros está seleccionando, ya que sobre estos es a quien se le realiza la acción de los botones. Ver figura 3.4.6

## Barra de Estatus



**Figura 3.4.7 Barra de estatus**

Su localización será por debajo del área de visualización de datos, en la parte inferior de la ventana principal. Tendrá una barra de progreso que indicará si la aplicación se encuentra realizando una consulta y su terminación. Además mostrará el nombre del usuario que se encuentra firmado en el sistema, la base de datos a la que apunta (es muy útil ya que se tendrán bases de datos de pruebas, por lo que es primordial mostrar sobre cual se está trabajando) y la versión del sistema. Ver figura 3.4.7

Se pretende que todos estos componentes en conjunto sirvan al usuario para tener un panorama claro sobre el trabajo que debe realizar. Un diseño aproximado de la ventana quedaría de la siguiente forma, ver figura 3.4.8:



NOMBRE DE LA APLICACIÓN

- X

MENÚ PRINCIPAL (MENÚ GENERAL + MENÚ PRINCIPAL DE LA APLICACIÓN)

MENÚ DE ACCESO RÁPIDO

MENÚ DE BÚSQUEDA

MENÚ DE FILTROS

ÁREA DE DATOS


### 3.5 Justificación de la metodología y software a utilizar

Para que un proyecto rinda los resultados esperados en el tiempo esperado y con los costos estimados y además de eso pudiéramos controlarlo de manera clara pensando siempre en la escalabilidad o en los nuevos requerimientos que el negocio pueda necesitar es necesario tomar en cuenta una de las muchas metodologías para el proceso de desarrollo de software.

Actualmente con el creciente desarrollo tecnológico y la aparición de nuevos modelos de producción, han ido apareciendo nuevas metodologías de proceso de desarrollo, estos procesos han ido tomando características relevantes, lo que ha conllevado en agruparlos en dos grandes grupos, los llamados "**métodos pesados**" y los "**métodos ligeros**", la diferencia más importante entre estos dos grandes grupos es que mientras los métodos pesados intentan obtener los resultados ayudándose principalmente de la documentación ordenada, los métodos ligeros tienen como base de sus resultados a la comunicación e interacción directa con todos los usuarios involucrados en el proceso.

#### **RUP (Proceso Racional Unificado)**

Este es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto aunque no sea de software, se basa en la documentación generada en cada uno de sus cuatro fases: 1. Inicio, 2. Elaboración (definición, análisis y diseño), 3. Construcción (implementación) y 4. Transición (fin del proyecto y puesta en producción) en las cuales se ejecutarán varias iteraciones (según el tamaño del proyecto).

RUP se basa en casos de uso para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, basándose en UML (Lenguaje de Modelado Unificado).

Para poder usar RUP antes hay que adaptarlo a las características de la empresa, y medir de manera exacta el tiempo, costos y todos los demás recursos involucrados en el proceso.

El RUP presenta 3 características que constituyen la esencia de todo el proceso de desarrollo:

1. Dirigido por los casos de uso.
2. Centrado en la arquitectura.
3. Ciclo de vida iterativo.

Otras características o ventajas de la aplicación de esta metodología son las siguientes:

- Reconoce que las necesidades del usuario y sus requerimientos no se pueden definir completamente al principio
- Permite evaluar tempranamente los riesgos en lugar de descubrir problemas en la integración final del sistema
- Reduce el costo del riesgo a los costos de un solo incremento
- Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan para obtener resultados claros a corto plazo
- Distribuye la carga de trabajo a lo largo del tiempo del proyecto ya que todas las disciplinas colaboran en cada iteración.
- Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño.

## **Otras metodologías: XP y MSF**

### **Metodología XP**

El Desarrollo extremo (eXtreme Programming) es una metodología de desarrollo que busca la simplicidad y ligereza en desarrollo de un proyecto de software; por esta razón esta metodología es apropiada para proyectos en los que durante su desarrollo pueden cambiar las necesidades del cliente y así mismo es necesario actualizar todo el proceso de ingeniería desarrollado; esta metodología busca adaptarse rápidamente al cambio disminuyendo todo lo posible los costes de: tiempo, talento humano, equipos, etc.

En esta metodología cambia un poco a la metodología RUP, la implementación de los casos de uso por las: Historias de usuario, las cuales básicamente son la descripción de las necesidades del cliente resaltando sus necesidades y los riesgos que estos generan. En base a estas historias se definen las tareas a desarrollar, las cuales se planifican en un tiempo por la prioridad que requiere en el proyecto, estas tareas son asignadas al grupo de desarrollo del proyecto.

De igual forma que en la metodología RUP, la comunicación dentro del grupo de trabajo es muy importante por ello es necesario llevar a cabo reuniones diarias de seguimiento y rotando las tareas asignadas con el fin de que todas las personas conozcan sobre el sistema, en los casos que suceden cuellos de botella dos programadores o más en conjunto desarrollan una tarea en un dolo equipo, con el fin de unificar esfuerzos en una sola tarea.

### **Metodología MSF**

El marco de trabajo de soluciones de Microsoft (Microsoft Solutions Framework), es una metodología que se puede implementar no solo en proyectos de software, sino también en proyectos de infraestructura, de redes, etc., esto se debe a que esta metodología no se centra en algún tipo de procedimiento ó paso a seguir , sino que es un “manual” de buenas prácticas (sin dejar a un lado la experiencia recolectada al

pasar de los proyectos desarrollados) a la hora de tomar decisiones, a la hora de saber que paso me conviene seguir.

Tal como las dos metodologías presentadas anteriormente (XP y RUP), aquí también es fundamental el trabajo donde cada miembro tiene claro sus funciones (tareas), manteniendo buena comunicación en el grupo con el fin de que todo el grupo esté al tanto del proyecto y todos mantengan un mismo nivel técnico, y así poder reaccionar a los cambios que puedan generarse durante el desarrollo del proyecto.

Debido a que esta metodología puede ser implementada en diferentes proyectos de TI, esta propone 2 modelos a seguir: el MSF Modelo de equipo y MSF Modelo de Gobierno, en una primera vista estos modelos siguen las mismas fases pero con otros nombres las cuales se resumen en: levantamiento de requerimientos, análisis, diseño, definición de arquitectura, desarrollo, pruebas, implementación y refactorización.

### Ventajas y desventajas

Metodología	Ventajas	Desventajas
RUP	<p>1 -Esta es una metodología completa en sí misma, con un énfasis en la documentación precisa</p> <p>2 -Es capaz de resolver de forma proactiva los riesgos del proyecto asociado a las nuevas necesidades de los clientes que requieren una cuidadosa gestión de solicitud de cambio</p> <p>3 -Se requiere menos tiempo para la integración en el proceso de integración va a lo largo del ciclo de vida de desarrollo.</p> <p>4 - El tiempo de desarrollo menor debido a la reutilización de componentes.</p>	<p>1 - Los miembros del equipo deben ser expertos en su campo para desarrollar un software bajo esta metodología.</p> <p>2-El proceso de desarrollo es demasiado complejo y desorganizado.</p> <p>3- En la reducción de los proyectos de vanguardia que utilizan las nuevas tecnologías, la reutilización de componentes no será posible</p> <p>4-La integración en el proceso de desarrollo de software, en teoría parece una buena cosa. Pero en particular los grandes proyectos de desarrollo con flujos múltiples que sólo servirá para aumentar la confusión y causar más problemas durante las etapas de la prueba</p>

XP	<ul style="list-style-type: none"> <li>1 - Programación organizada.</li> <li>2 - Menor tasa de errores.</li> <li>3 - Satisfacción del programador.</li> <li>4 - Presto a cambios</li> </ul>	<ul style="list-style-type: none"> <li>1- Es factible solo en proyectos a corto plazo.</li> <li>2- La falta de documentación que nos e hace, ya que todos saben del proyecto, no permite recoger la experiencia para próximos proyectos</li> </ul>
MSF	<ul style="list-style-type: none"> <li>1- Abarca las tareas de riesgos.</li> <li>2 - Apoya el desarrollo basado en componentes (nuevos o existentes).</li> <li>3 - Se puede utilizar en proyectos grandes y pequeños.</li> <li>4 - Asegura la calidad.</li> </ul>	<ul style="list-style-type: none"> <li>1 -Tiene las desventajas del modelo espiral debido a las iteraciones en cada ciclo y puede tomar mucho más tiempo.</li> <li>2 - Por el grado de complejidad puede no resultar muy adecuado.</li> <li>3 - Esta más enfocado a la administración de proyecto</li> <li>4- Poca documentación del proceso</li> </ul>

**Figura 3.5.1 Comparación entre las metodologías RUP, XP y MSF**

Realizando comparaciones con la información encontrada, comentarios de casos de éxito en algunas empresas, la madurez que ha tenido cada metodología al pasar del tiempo; creemos que es muy importante una metodología que ayude a soportar el avance de un proyecto tanto con hechos como con registro de los hechos, es decir una metodología que permita llegar al producto software requerido con la documentación técnica que lo soporte, con el fin de fortalecer la experiencia para próximos desarrollos; por esta razón creemos que la metodología XP no es la más apropiada para apoyar esa parte del proyecto; es así que queda por evaluar la metodología RUP y la metodología MSF; estas dos metodologías poseen un fuerte enfoque hacia la calidad del producto software, debido al trabajo iterativo y conceptos de refinamiento que poseen, a pesar que poseen este mismo enfoque son metodologías diferentes, la metodología MSF está más centrada a la administración del proyecto, mientras que RUP se centra más al proceso del proyecto desde diferentes vistas, teniendo en cuenta el equipo de trabajo, duración del proyecto, complejidad, etc., además que la documentación de MSF es muy escasa.

Es así que concluimos que la metodología más adecuada para el desarrollo de este proyecto es la metodología RUP.

### **Metodología UML:**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

## Tablas comparativas entre SDBD

Vea las figuras 3.5.2, 3.5.3, 3.5.4, 3.5.5 y 3.5.6

### CARACTERISTICAS GENERALES

	MS SQL	ORACLE	DB2	MySQL
Desarrollo de base de datos	X	X	X	
Herramientas de consulta y ajustes de las base de datos	X	X	X	
Administrador corporativo	X			

Figura 3.5.2 Características generales

### VARIANTES Y EXTENSIONES DE SQL

	MS SQL	ORACLE	DB2	MySQL
Tipos de datos	X	X	X	
Vistas	X	X		
Vistas actualizadas	X			
Desencadenadores (disparadores)	X	X	X	
Connect by		X		
Upsert e inserciones en varias tablas		X		
Clausula with	X	X	X	
Funciones y métodos definidos por el usuario	X	X	X	
Soporte para XML	X	X	X	
Extensiones de índices y restricciones	X	X	X	

Figura 3.5.3 Variantes y extensiones de sql

ALMACENAMIENTO E INDEXACIÓN

	MS SQL	ORACLE	DB2	MySQL
Tablas	X	X	X	
Índices	X	X	X	
Registros	X	X	X	
Arquitectura de almacenamiento (administración y segmentación)	X	X	X	
Grupos de archivos	X	X	X	

**Figura 3.5.4 Almacenamiento e indexación**

PROCESAMIENTO Y OPTIMIZACIÓN DE CONSULTAS

	MS SQL	ORACLE	DB2	MySQL
Métodos de acceso	X	X	X	
Optimización de consultas	X	X	X	
Simplificación de consultas	X	X	X	
Mezcla de vistas		X		
Operaciones	X	X	X	
Reordenación y optimización basada en el coste	X			

**Figura 3.5.5 Procesamiento y optimización de consultas**

CONTROL DE CONCURRENCIA Y RECUPERACIÓN

	MS SQL	ORACLE	DB2	MySQL
Concurrencia y aislamiento	X		X	
Compromiso y retroceso			X	
Registro histórico y recuperación	X	X	X	
Transacciones	X	X	X	
Bloqueos	X	X	X	
Estructuras básicas de recuperación	X	X	X	
Bases de datos en espera gestionadas		X		

**Figura 3.5.6 Control de concurrencia y recuperación**

## Enterprise Library

Microsoft Enterprise Library es un conjunto de componentes de software reusables conocidos como Application Blocks, diseñado para ayudar a los desarrolladores de software en los aspectos transversales y verticales de un sistema, como son:

- El registro de errores o sucesos
- Validaciones de datos y objetos
- Acceso a datos
- Manejos de excepciones
- Criptografía
- Configuración
- Seguridad
- Inyección de dependencias

El diseño de los *Application Blocks*, encapsula prácticas recomendadas y probadas para desarrollos *.Net*. Esas buenas prácticas están en el diseño de la *Enterprise Library*, así como en los ejemplos de inicio rápido que acompañan la documentación de estas bibliotecas. Los programadores pueden agregar los *Application Blocks* de una manera sencilla. Por ejemplo el *Application Block* de Acceso a datos, incorpora todas las funcionalidades más comunes de ADO.NET, exponiéndolas a través de clases que son muy fáciles de emplear.

Los bloques que se emplean para el desarrollo son:

- Acceso a datos
- Manejo de Excepciones
- Registro de errores y sucesos
- Validaciones de datos y objetos

## **DevExpress**

Los *DevExpress* para Windows Forms de *.Net* son componentes principalmente gráficos que nos permiten lograr un mejor aspecto y navegabilidad de nuestra aplicación, ya que proveen controles con amplias capacidades de manipulación para el programador. En un principio pueden parecer abrumadores ya que tienen bastantes propiedades y métodos con los que se puede interactuar, lo que genera una curva de aprendizaje, sin embargo su documentación y ayuda en línea es muy sólida, motivo por el cual se eligieron para desarrollar la aplicación. Uno de los principales motivos de elección de la suite de *DevExpress*, fue por la experiencia que ya tenía uno de los miembros del equipo en trabajar con ellos, lo que representaba una ventaja más sobre los demás controles.

## **Microsoft Visual C Sharp**

Microsoft Visual C# 2008 es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C.

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

Algunas de las ventajas de utilizar Visual C# con SQL Server:

- **Modelo de programación mejorada:** los lenguajes de .NET Framework proporcionan construcciones y funcionalidades previamente no disponibles para los desarrolladores de SQL Server.
- **Seguridad mejorada:** el código administrado ejecuta un entorno de Common Language Runtime que se hospeda en el motor de la base de datos. De esta forma, los objetos de base de datos de .NET Framework están más seguros que los procedimientos almacenados extendidos disponibles en versiones anteriores de SQL Server. Al usar objetos de base de datos creados en Visual C#, la seguridad de acceso del código de esos lenguajes se combina con los permisos basados en usuario de SQL Server.
- **Tipos y agregados definidos por el usuario** los tipos y agregados definidos por el usuario son dos objetos de base de datos administrados que expanden la capacidad de almacenamiento y consulta de SQL Server.
- **Entorno de desarrollo común:** el desarrollo de la base de datos se integra al entorno de desarrollo de Microsoft Visual Studio. Los desarrolladores utilizan las mismas herramientas para desarrollar y depurar objetos de base de datos y scripts que las que usan para escribir componentes y servicios de .NET Framework de nivel medio o nivel de cliente.
- **Mejor rendimiento:** algunas funciones, como las que ejecutan operaciones matemáticas en cada fila de una base de datos, pueden ofrecer un mejor rendimiento si son ensamblados compilados a partir de un proyecto de Visual C#.
- **Riqueza del lenguaje** Visual C# proporciona funcionalidad que no está disponible en otros lenguajes como matrices, control de excepciones sofisticado y reusabilidad de código.
- **Extensibilidad** mediante Visual C# puede crear dos objetos de base de datos que no se pueden crear en otros lenguajes: agregados y tipos definidos por el usuario.
- **Utilización de conocimientos existentes:** es posible utilizar y mejorar sus conocimientos en los lenguajes y desarrollar entornos en los que ya existe experiencia para crear objetos de base de datos.

- **Experiencia del programador más enriquecida:** al desarrollar objetos de base de datos mediante la plantilla de proyecto de SQL Server, se cuenta con integración completa con el sistema de proyectos. Esto incluye compilación, depuración e implementación a varios servidores.

**Estabilidad y confiabilidad:** los objetos de base de datos que se crean con Visual C# son seguros, estables, sólidos y confiables. Cuando se ejecutan procedimientos almacenados creados en Visual C#, el procedimiento almacenado no se ocupa de la administración de memoria ni del subprocesamiento y, por lo tanto, se controlan de manera más sólida.

## **CAPÍTULO IV. “DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN”**

### **4.1 El modelado del sistema**

Para producir software que cumpla su propósito, los requisitos del sistema deben estar claros, con este fin se trabaja activamente con los usuarios haciéndolos participar para que no queden requerimientos sin atender. Para conseguir un software de calidad, que sea duradero y fácil de mantener hay que idear una sólida base arquitectónica que sea flexible al cambio. Para desarrollar software de forma rápida y eficientemente, minimizando el trabajo de recodificación y evitando la creación de código inútil, hay que disponer, además de la gente y las herramientas necesarias, de un enfoque apropiado.

Para conseguir, desarrollar software de manera profesional y se obtenga un producto de calidad, es completamente necesario seguir una metodología y no abordar los problemas de manera somera, con el fin de obtener un modelo que represente bien la solución al problema. El modelado es la espina dorsal del desarrollo del software de calidad. Se construyen modelos para explicar el comportamiento del sistema a desarrollar, para comprender mejor ese sistema, para controlar el riesgo y en definitiva para poder atacar problemas que sin el modelado su resolución sería imposible, tanto desde el punto de vista de los desarrolladores (no se pueden cumplir los plazos estimados, no se consigue ajustar los presupuestos...) como desde el punto de vista del cliente, el cual, si finalmente se le entrega el producto del desarrollo, se encontrará con infinidad de problemas, desde que no se cumplen las especificaciones hasta fallos que dejan inutilizado el sistema.

El modelo proporciona los planos de un sistema, desde los más generales, que proporcionan una visión global del sistema, hasta los más detallados. En un modelo se han de incluir los elementos que tengan más relevancia y omitir los que no son

interesantes para el nivel de abstracción que se ha elegido. A través del modelado se plantearán los siguientes objetivos:

- Ayudan a visualizar cómo es o como será el sistema.
- Especificar la estructura o el comportamiento de un sistema.
- Obtener las plantillas que guían en la construcción de un sistema.
- Documentar las decisiones adoptadas.

Un modelo representa a un sistema de software desde una perspectiva específica. Cada diagrama permite observar en un aspecto distinto el sistema. Por lo tanto nuestro sistema estará documentado por medio de los diagramas que proporciona el Lenguaje de Modelado Unificado (UML) y basado en la Metodología RUP.

Los diagramas de UML con los que vamos a trabajar son los siguientes:

- Diagrama de Casos de Uso.
- Diagrama de entidad - relación.
- Diagrama de Clases.
- Diccionario de datos.

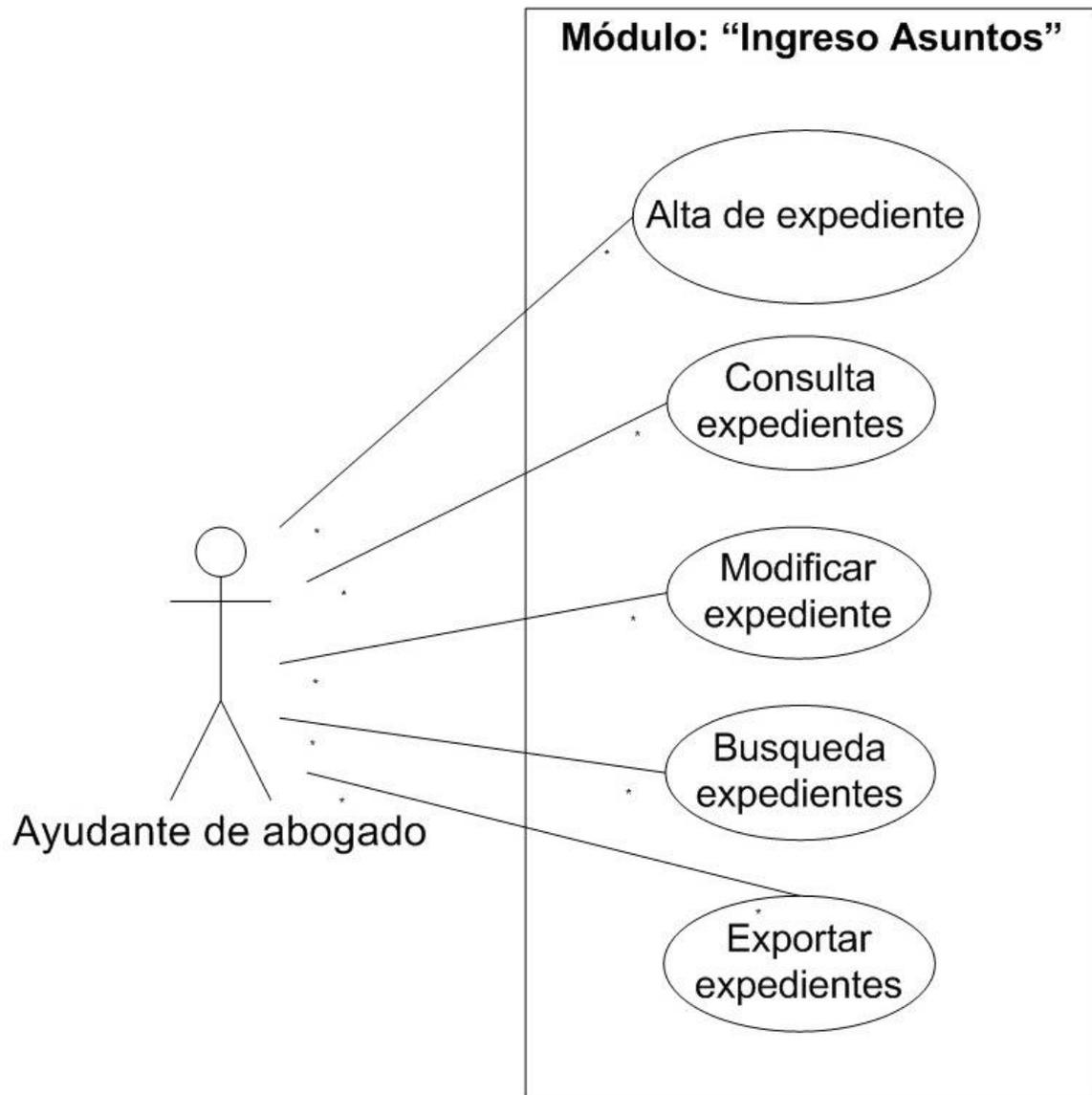
#### **4.1.1 Diagrama de casos de uso**

Un diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Nuestro sistema consta de 3 módulos que interactúan entre ellos y que a su vez son independientes uno de otro en su funcionamiento y programación. En las siguientes figuras se mostrarán los casos de uso de cada uno de estos módulos, respetando el flujo de acción de nuestro sistema:

- Módulo de Ingreso de Asuntos

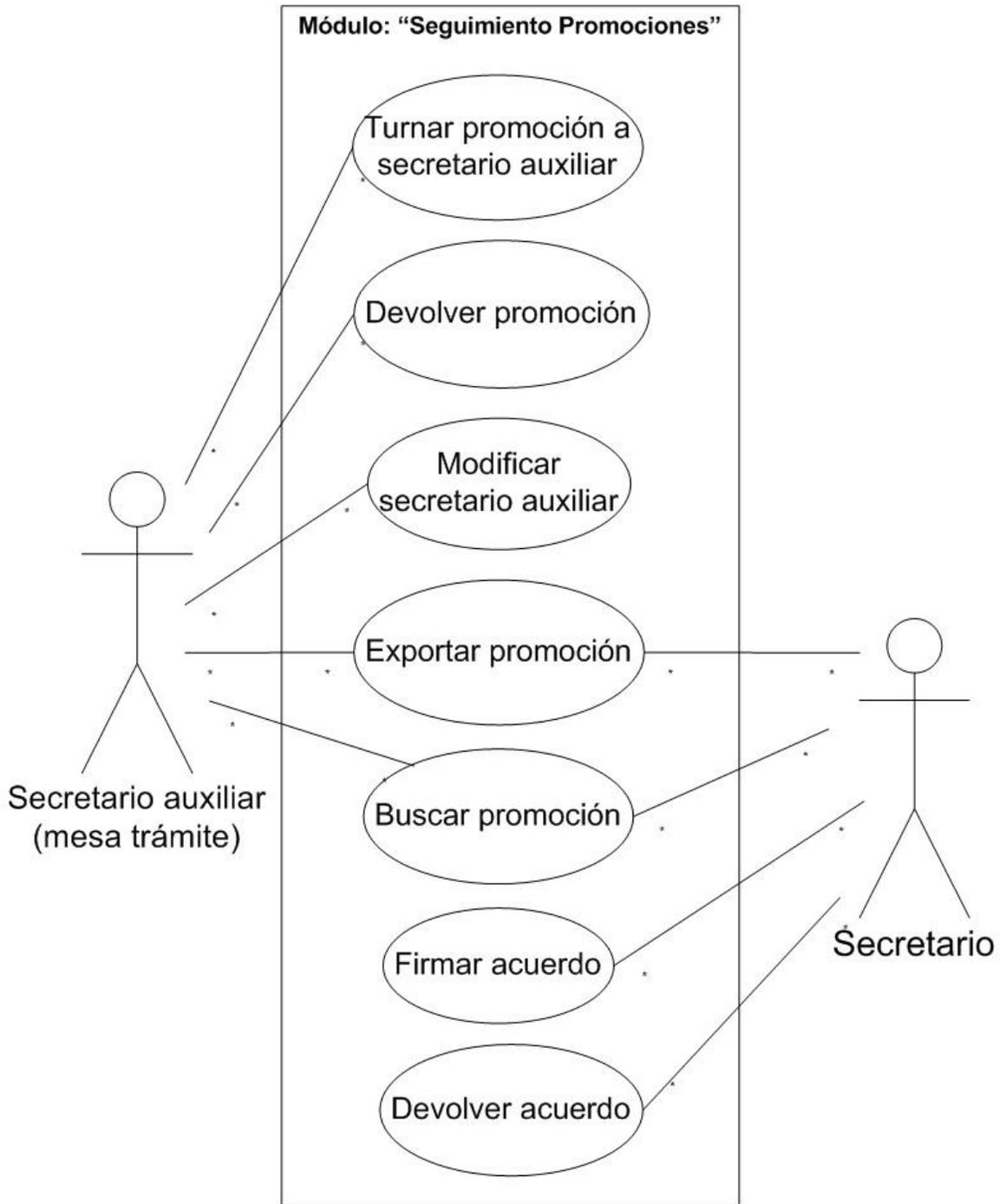
- Módulo de Seguimiento de Promociones
- Módulo de Acuerdos

El módulo de Ingreso de asuntos es el primer paso en el flujo de información de nuestro sistema y la figura 4.1.1.1 representa sus casos de uso.



**Figura 4.1.1.1 Diagrama de casos de uso para el módulo "Ingreso Asuntos"**

El siguiente módulo en este flujo de información es el de “Seguimiento de Promociones” y su diagrama de casos de uso está representado en la figura 4.1.1.2 que se muestra a continuación.



**Figura 4.1.1.2 Diagrama de casos de uso del módulo “Seguimiento de Promociones”**

El tercer y último módulo es el de “Acuerdos” y la figura 4.1.1.3 representa sus casos de uso.

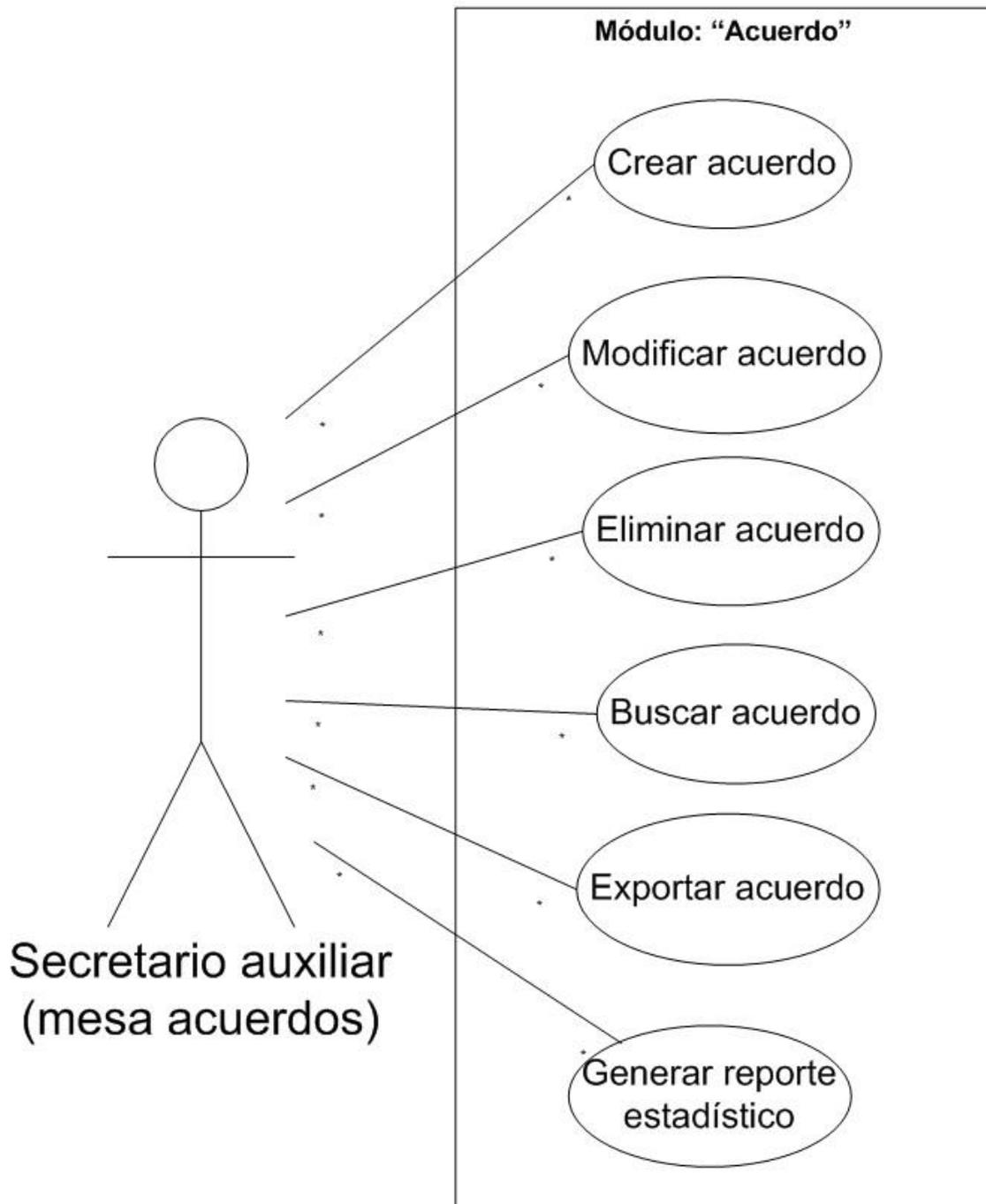


Figura 4.1.1.3 Diagrama de casos de uso del módulo “Acuerdos”

#### 4.1.2 Diagrama de modelado de datos

Un modelo de datos es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. Las características principales de un modelo de datos son:

- Analizar los aspectos de interés para una organización y la relación que tienen unos con otros.
- Descubrir y documentar los recursos de datos del negocio.
- Registra los requerimientos de datos de un proceso de negocio.

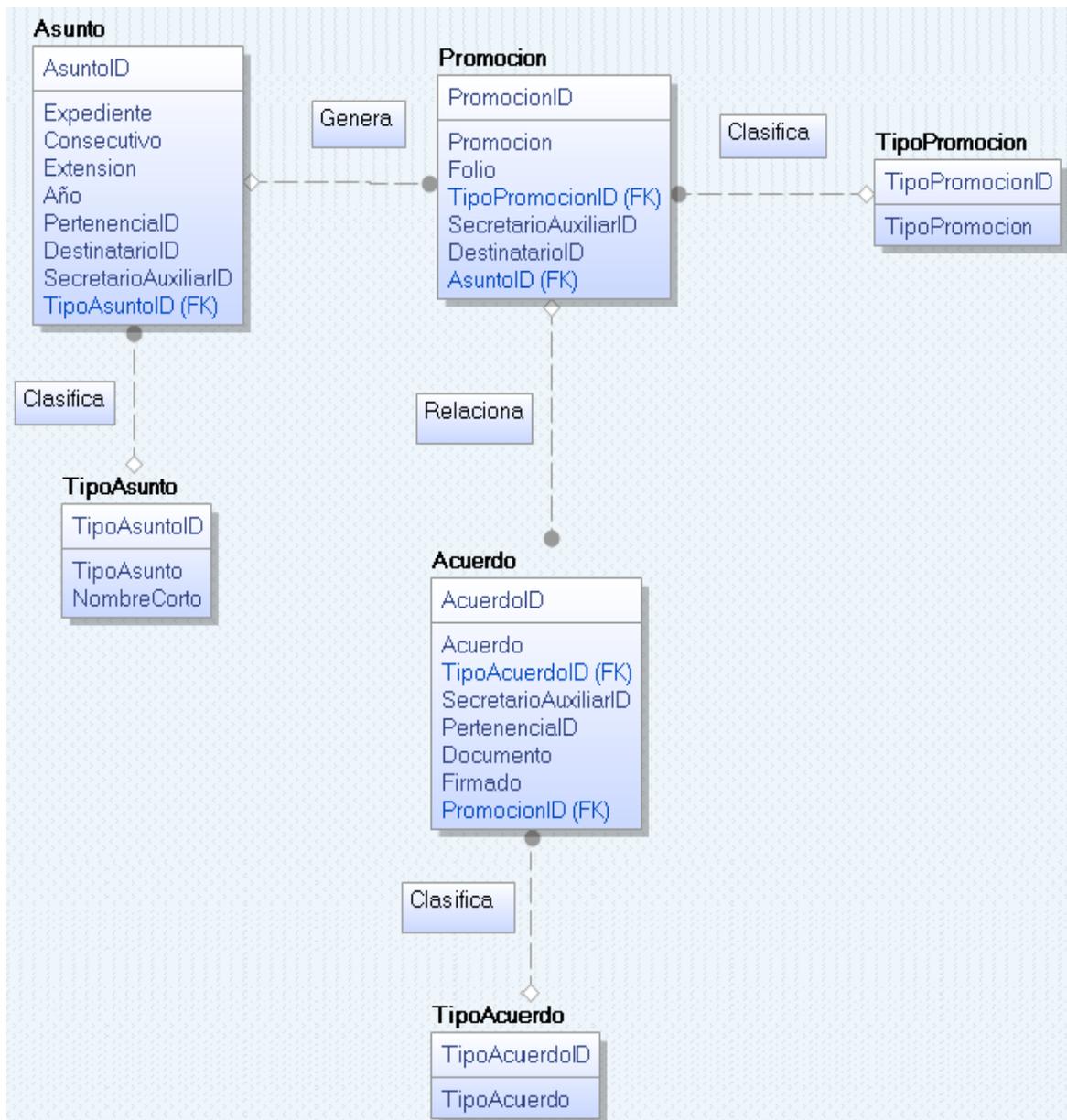
Tipos de modelado de datos básicamente son 3:

- **Conceptual:** muy general y abstracto, visión general del negocio/institución.
- **Lógico:** versión completa que incluye todos los detalles acerca de los datos.
- **Físico:** esquema que se implementara en un manejador de bases de datos (DBMS).

En la figura 4.1.2.1 se muestra el diagrama de modelado de datos sobre el flujo de un expediente.

El ciclo de vida de un expediente se inicia dando de alta un registro en la tabla Asunto, con los aspectos más generales y una clasificación que define el usuario. Cuando se da de alta un expediente se genera automáticamente un registro en la tabla Promocion, la cual contiene aspectos más detallados.

El usuario elige a qué secretario auxiliar será turnada dicha promoción, para su oportuno análisis. Después de este paso, el secretario genera el registro en la tabla Acuerdo anexando un documento que justifica su resolución. El último paso es que otro secretario auxiliar coloque el estatus de firmado al acuerdo, dándole así el visto bueno y la finalización del expediente.



**Figura 4.1.2.1 Modelado de datos para el flujo de un expediente**

La figura 4.1.2.2 muestra el diagrama de modelado de datos sobre el usuario que utilizará el sistema.

Se crearán varios perfiles en base a las operaciones que realizarán los usuarios, pudiendo así restringir operaciones críticas o búsqueda de información sensible. La tabla Usuario contendrá información como el login y password para ingresar al

sistema, el área y órgano de radicación, y muy importante también el secretario auxiliar que tiene asignado. En la tabla Operacion se tendrán todas las acciones que se podrán realizar sobre los tres módulos que se desarrollaron, los cuales tendrán filtros predeterminados guardados en la tabla FiltroSistema.

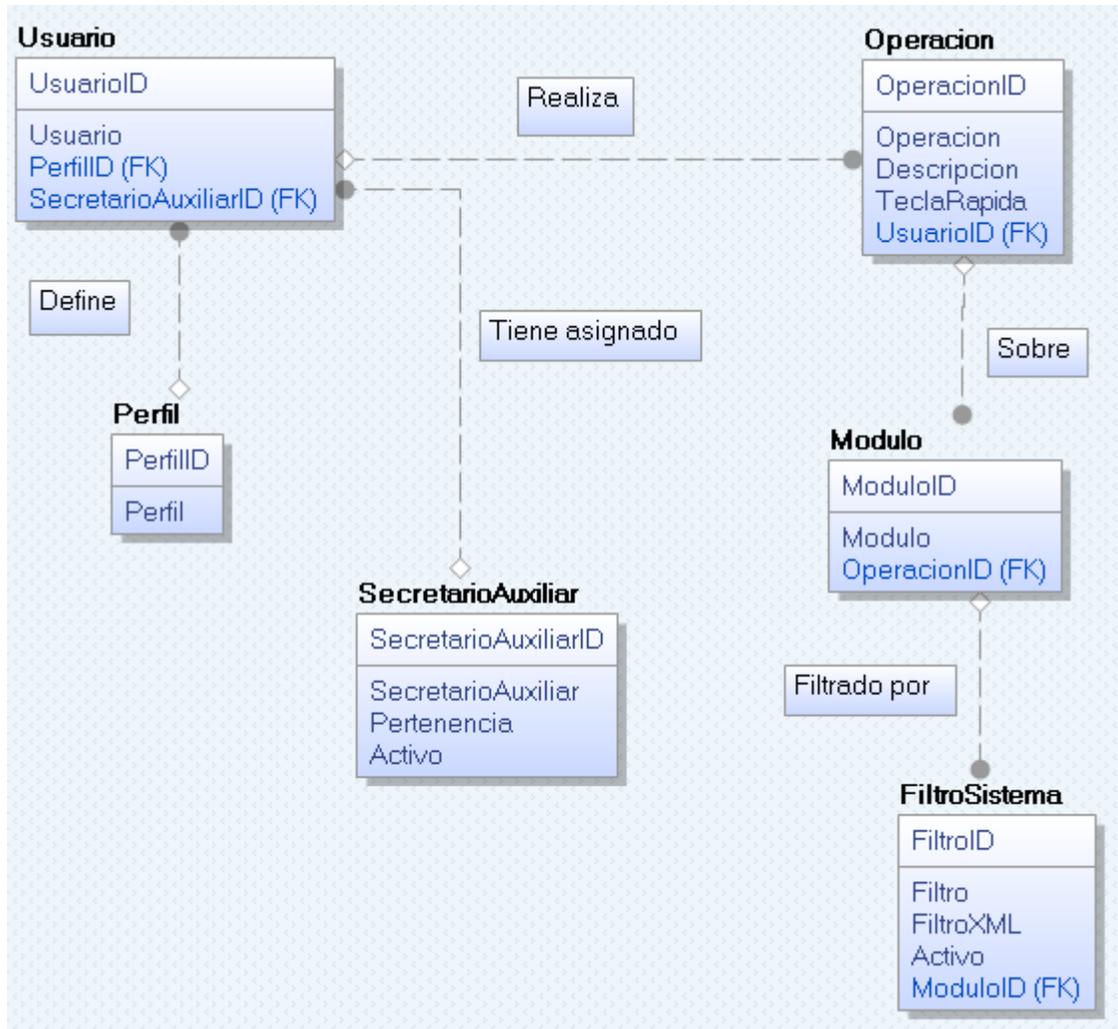


Figura 4.1.2.2 Modelado de datos para el usuario

### 4.1.3 Diagrama de Entidad – Relación

Generalmente todo modelo tiene una representación gráfica, para el caso de datos el modelo más utilizado es el modelo entidad-relación o diagrama E-R.

Se denomina así debido a que precisamente permite representar relaciones entre entidades (objetivo del modelado de datos).

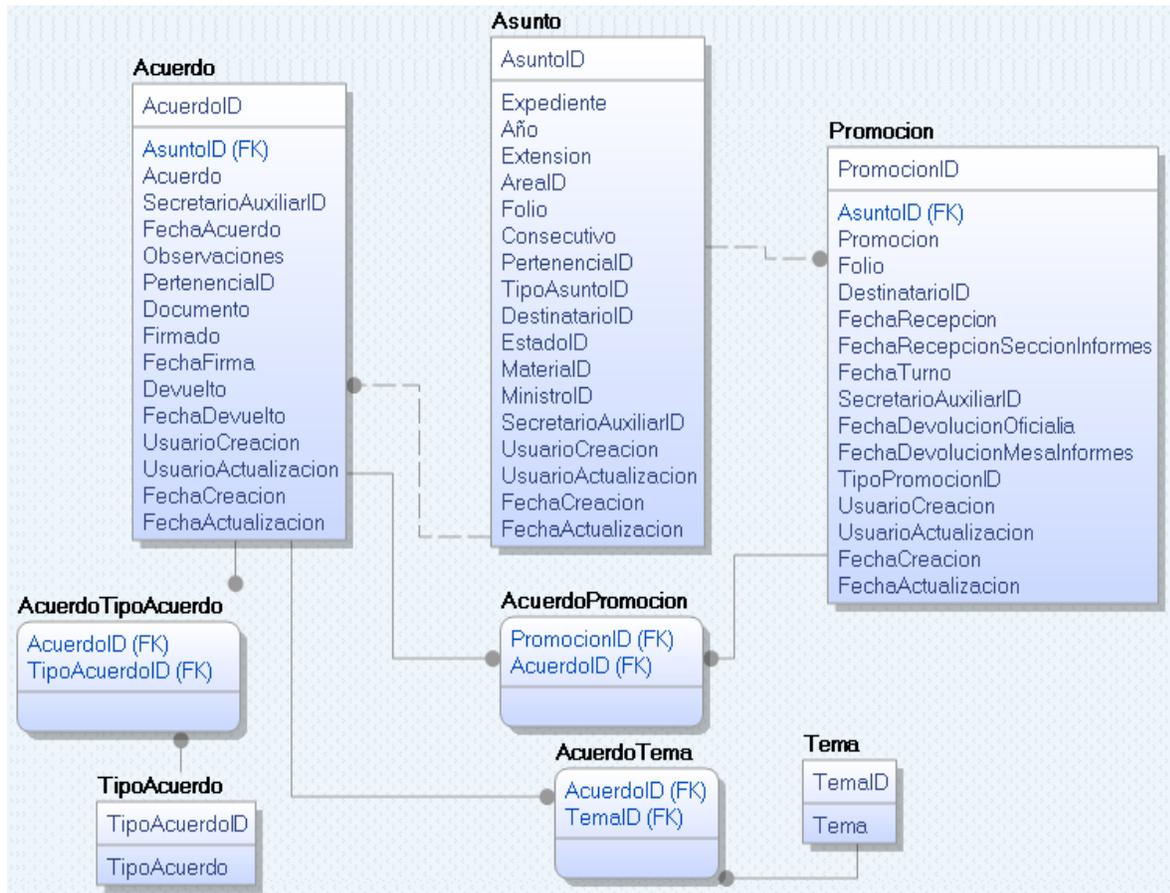
El modelo entidad-relación se compone por:

- **Entidades:** Todo lo que existe y es capaz de ser descrito (sustantivo).
- **Atributos:** Es una característica (adjetivo) de una entidad que puede hacer una de tres cosas:
  - Identificar
  - Relacionar
  - Describir
  
- **Relaciones:** La conexión que existe entre 2 entidades (verbo).
- **Cardinalidad:** Número de ocurrencias que pueden existir entre un par de entidades.
- **Súper llave:** Conjunto de uno o más atributos que "juntos" identifican de manera única a una entidad
- **Llave candidata:** Es una súper llave mínima
- **Llave primaria:** La seleccionada para identificar a los elementos de un conjunto de entidades.

En base al análisis del modelado de datos del punto anterior, podemos crear el diagrama Entidad-Relación de algunas tablas que intervienen con el expediente.

Debido a que los acuerdos pueden tener más de una clasificación, se crea una tabla AcuerdoTipoAcuerdo donde se guarda ésta relación. Es el mismo caso para la tabla

de AcuerdoTema. La tabla principal de toda la base será Asunto, desde ahí comienza el ciclo de vida de un expediente y se relacionan las demás tablas. Ver figura 4.1.3.1



**Figura 4.1.3.1 Diagrama Entidad-Relación sobre expedientes**

Por el lado de los usuarios, debido a que el mismo usuario puede desempeñar tareas de distintos módulos, es necesario crear una tabla PerfilUsuario para poder asignarle los perfiles que sean necesarios. Pasa lo mismo con los secretarios auxiliares, el mismo usuario podrá tener asignados más de uno, por lo que se crea la tabla UsuarioSecretarioAuxiliar para tal propósito. Cada perfil tendrá relacionadas operaciones diferentes, pudiendo ser éstas de diferentes módulos, la tabla PerfilOperacion guardará ésta relación. Ver figura 4.1.3.2

Se tendrá una tabla llamada SystemColumns donde se guardarán las columnas que mostrará cada módulo, y para los filtros se tendrá la tabla FiltroSistema, igual relacionados a cada módulo. La relación entre el usuario y los módulos se tendrá mediante UsuarioFiltroSistema, donde se le asignará a cada usuario los filtros a los que tendrá acceso, y por ende, a los módulos que podrá consultar.

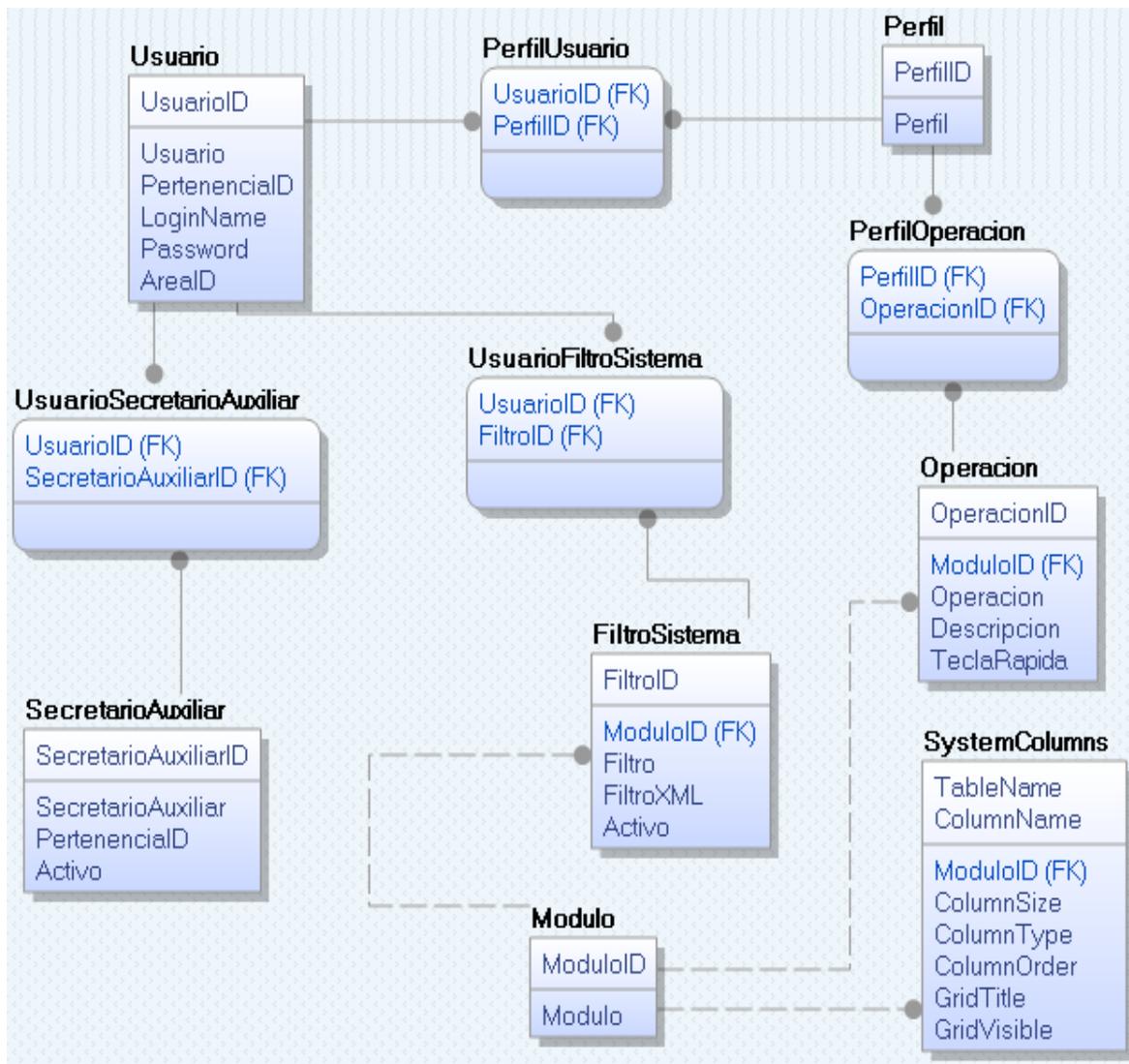


Figura 4.1.3.2 Diagrama Entidad-Relación sobre el usuario

#### 4.1.4 Diagrama de clases

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir, los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

La clase define el ámbito de definición de un conjunto de objetos. Cada objeto pertenece a una clase. Los objetos se crean por instanciación de las clases.

Cada clase se representa en un rectángulo con tres compartimientos:

- Nombre de la clase
- Atributos de la clase
- Operaciones de la clase

Los atributos de las clases son los siguientes:

- **Tipo:** puede llegar a depender del lenguaje de programación a utilizar.
- **Valor inicial:** valor que poseerá el atributo al crear un objeto.
- **Visibilidad:** está relacionado con el encapsulamiento.
- **Multiplicidad:** determinar si un atributo debe estar o no, y si posee un único valor o una lista de valores.
- **Ordenamiento:** especifica si el atributo determina alguna relación de orden dentro de la clase.
- **Capacidad de cambio:** permite definir atributos con valores constantes.
- **Modificadores:** un atributo puede ser de clase, derivado, volátil, transitorio.

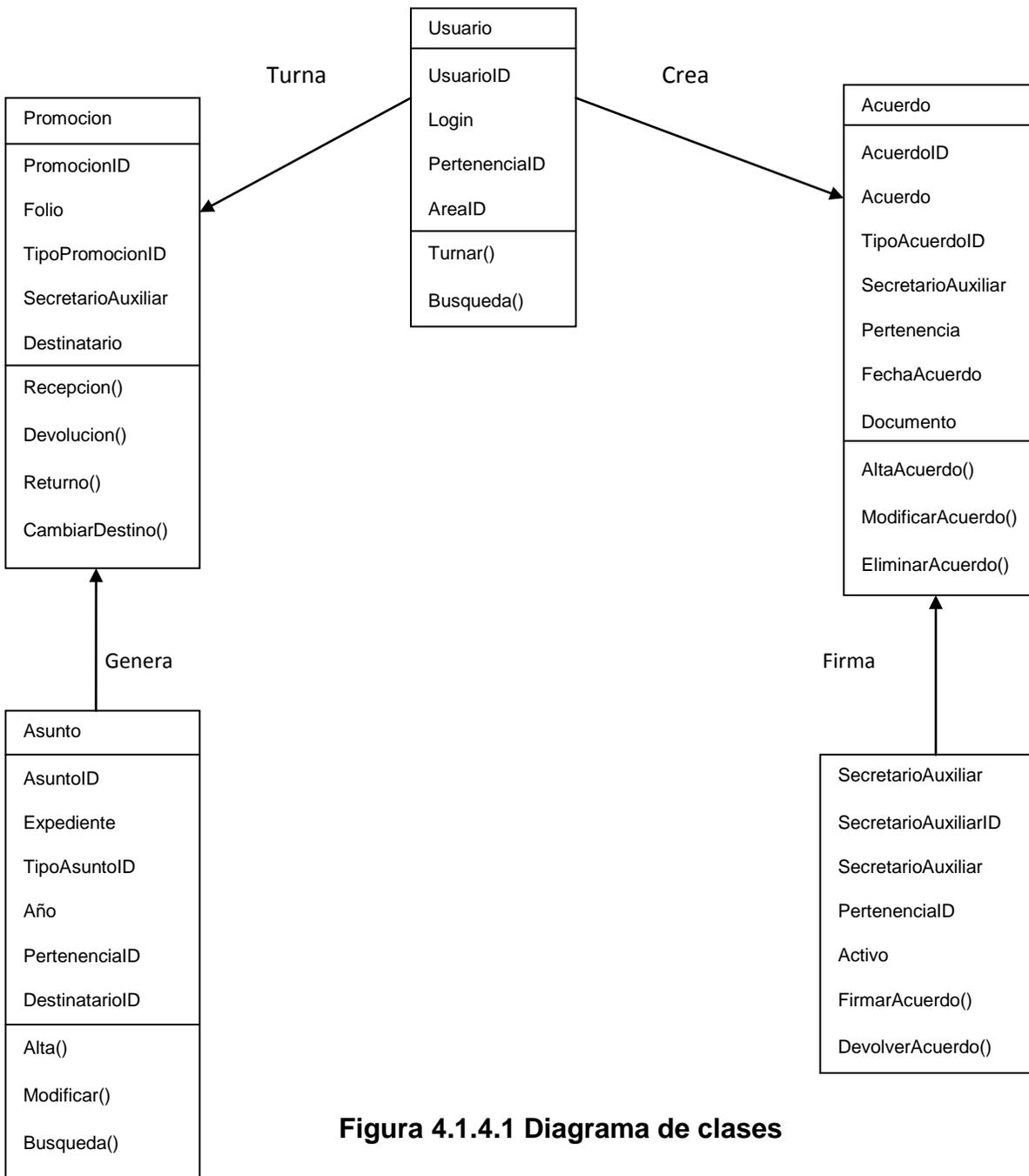
Niveles de encapsulamiento:

(-) Privado: es el más fuerte. Esta parte es totalmente invisible desde fuera de la clase.

(#) Los atributos/operaciones protegidos están visibles para las clases amigas y para las clases derivadas de la original.

(+) Los atributos/operaciones públicos son visibles a otras clases.

En la figura 4.1.4.1 se muestra el diagrama de clases del flujo que sigue un expediente:



**Figura 4.1.4.1 Diagrama de clases**

### 4.1.5 Diccionario de Datos

Un diccionario de datos es un conjunto de registros que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En el diccionario de datos se documenta la información generada por el diseñador de bases de datos, sus principales utilidades son:

- Describir el propósito de la Base de Datos.
- Descripción detallada de cada tabla dentro de la Base de Datos
- Documentar la estructura interna de cada tabla, incluyendo sus campos y sus tipos de datos con comentarios, todos los índices y todas las vistas.
- Describir reglas, como pueden ser el uso de valores no nulos o valores únicos.

En las siguientes figuras se representarán las tablas que se utilizarán en nuestro sistema. Ver tablas 4.1.5.1 hasta la 4.1.5.19

Tabla Modulo						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID Módulo	ModuloID	Int (4)	PK	No		Identificador del módulo
Módulo	Modulo	Varchar (256)		Sí		Nombre del módulo

**Tabla 4.1.5.1 Modulo**

Tabla SystemColumns						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
Nombre de la tabla	TableName	Varchar (50)	PK	No		Nombre para tabla
Nombre de la columna	ColumnName	Varchar (50)	PK	No		Nombre para columna
ID Módulo	ModuloID	Int (4)	FK	Sí	Modulo	Identificador del módulo
Tamaño de columna	ColumnSize	Int (4)		Sí		Tamaño de la columna
Tipo de columna	ColumnType	Varchar (20)		Sí		Tipo de la columna
Orden de columna	ColumnOrder	Int (4)		Sí		Orden de la columna
Título del grid	GridTitle	Varchar (50)		Sí		Título del grid
Grid visible	GridVisible	Bit (1)		Sí		Define si el grid es visible o no

**Tabla 4.1.5.2 SystemColumns**

Tabla FiltroSistema						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID Filtro	FiltroID	Int (4)	PK	No		Identificador del Filtro
ID Módulo	ModuloID	Int (4)	FK	Sí	Modulo	Identificador del módulo
Archivo XML del filtro	FiltroXML	Xml (.)		Sí		XML asociado al filtro
Filtro activo	Activo	Bit (1)		Sí		Define si está activo o no

**Tabla 4.1.5.3 FiltroSistema**

Tabla Operacion						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
<b>ID operación</b>	OperacionID	Int (4)	PK	No		Identificador de la operación
<b>ID Módulo</b>	ModuloID	Int (4)	FK	Sí	Modulo	Identificador del módulo
<b>Nombre de la operación</b>	Operacion	Varchar (50)		Sí		Nombre de la operación
<b>Descripción</b>	Descripcion	Varchar (100)		Sí		Descripción de la operación
<b>Tecla rápida</b>	TeclaRapida	Varchar (20)		Sí		Texto para la tecla rápida

**Tabla 4.1.5.4 Operacion**

Tabla Perfil						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
<b>ID perfil</b>	PerfilID	Int (4)	PK	No		Identificador del perfil
<b>Nombre perfil</b>	Perfil	Varchar (100)		Sí		Nombre para el perfil

**Tabla 4.1.5.5 Perfil**

Tabla PerfilOperacion						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
<b>ID perfil</b>	PerfilID	Int (4)	FK	No	Perfil	Identificador del perfil
<b>ID operación</b>	OperacionID	Int (4)	FK	No	Operacion	Identificador de la operación

**Tabla 4.1.5.6 PerfilOperacion**

Tabla Usuario						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID usuario	UsuariolD	Int (4)	PK	No		Identificador de usuario
Usuario	Usuario	Varchar (100)		No		Nombre del usuario
Login	Login	Varchar (50)		No		Nombre de usuario login
Password	Password	Varchar (50)		Sí		Password del usuario
ID pertenencia	PertenencialD	Int (4)		Sí		Sección a la que pertenece
ID Área	AreaID	Int (4)		Sí		Área a la que pertenece

**Tabla 4.1.5.7 Usuario**

Tabla PerfilUsuario						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID usuario	UsuariolD	Int (4)	FK	No	Usuario	Identificador de usuario
ID perfil	PerfillD	Int (4)	FK	No	Perfil	Identificador del perfil

**Tabla 4.1.5.8 PerfilUsuario**

Tabla UsuarioFiltroSistema						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID usuario	UsuariolD	Int (4)	FK	No	Usuario	Identificador de usuario
ID Filtro	FiltroID	Int (4)	FK	No	Filtro	Identificador del Filtro

**Tabla 4.1.5.9 UsuarioFiltroSistema**

Tabla SecretarioAuxiliar						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID S.A.	SecretarioAuxiliarID	Int (4)	PK	No		Identificador del secretario aux.
S.A.	SecretarioAuxiliar	Varchar (50)		No		Nombre del secretario
Sector	PertenenciaID	Int (4)		No		Área a la que pertenece
Activo	Activo	Bit (1)		No		Define si está activo o no

**Tabla 4.1.5.10 SecretarioAuxiliar**

Tabla UsuarioSecretarioAuxiliar						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID usuario	UsuariID	Int (4)	FK	No	Usuario	Identificador de usuario
ID S.A.	SecretarioAuxiliarID	Int (4)	PK	No		Identificador del secretario aux.

**Tabla 4.1.5.11 UsuarioSecretarioAuxiliar**

Tabla Tema						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID Tema	TemaID	Int (4)	PK	No		Identificador del tema
Nombre tema	Tema	Varchar (300)		No		Nombre del tema ingresado

**Tabla 4.1.5.12 Tema**

Tabla TipoAcuerdo						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID tipo de acuerdo	TipoAcuerdoID	Int (4)	PK	No		Identificador del tipo de acuerdo
Nombre tipo de acuerdo	TipoAcuerdo	Varchar (100)		No		Nombre del tipo de acuerdo

**Tabla 4.1.5.13 TipoAcuerdo**

Tabla Asunto						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
ID de asunto	AsuntoID	Int (4)	PK	No		Identificador del asunto
Expediente asunto	Expediente	Varchar (15)		No		Descripción del asunto
Año asunto	Anio	Int (4)		No		Año del asunto
Extensión asunto	Extension	Char (2)		No		Extensión de asunto
ID de área creación	AreaID	Int (4)		Sí		Identificador del área que lo crea
Folio asunto	Folio	Varchar (20)		Sí		Folio del asunto
Número consecutivo creación	Consecutivo	Int (4)		No		Número consecutivo de asunto creado
Nombre del dueño	PertenencialID	Int (4)		Sí		Identificador de persona a la que pertenece
ID tipo asunto	TipoAsuntoID	Int (4)		No		Identificador del tipo de asunto
ID al que va dirigido	DestinatarioID	Int (4)		Sí		Identificador de la persona a la que va dirigido
ID del estado de	EstadoID	Int (4)		Sí		Identificador del estado de

<b>procedencia</b>						procedencia del asunto
<b>ID materia asunto</b>	MaterialID	Int (4)		Sí		Identificador de la materia del asunto
<b>ID ministro pertenencia</b>	MinistroID	Int (4)		Sí		Identificador del ministro al que le pertenece
<b>ID Secretario Auxiliar</b>	SecretarioAuxiliarID	Int (4)		Sí		Identificador del secretario al que le pertenece
<b>Nombre usuario</b>	UsuarioCreacion	Varchar (50)		No		Nombre del usuario que creó el asunto
<b>Nombre usuario actualización</b>	UsuarioActualizacion	Varchar (50)		Sí		Nombre del usuario que actualiza asunto
<b>Fecha creación asunto</b>	FechaCreacion	Datetime		No		Fecha de creación del asunto
<b>Fecha actualización asunto</b>	FechaActualizacion	Datetime		Sí		Fecha de última actualización del asunto

**Tabla 4.1.5.14 Asunto**

<b>Tabla Promocion</b>						
<b>Nombre</b>	<b>Acrónimo</b>	<b>Tipo y Longitud</b>	<b>Llave</b>	<b>Nulos</b>	<b>Tablas Asociadas</b>	<b>Descripción</b>
<b>ID Promoción</b>	PromocionID	Int (4)	PK	No		Identificador de la promoción
<b>ID Asunto</b>	AsuntoID	Int(4)	FK	No	Asunto	Identificador del asunto
<b>Asunto de promoción</b>	Promocion	Varchar (max)		Sí		Nombre identificador de la promoción

<b>Folio promoción</b>	Folio	Varchar (20)		Sí		Folio de la promoción
<b>ID destinatario</b>	DestinatarioID	Int (4)		Sí		Identificador del destinatario
<b>Fecha recepción</b>	FechaRecepcion	Datetime		Sí		Fecha de la recepción del asunto
<b>Fecha recepción de informe</b>	FechaRecepcion SeccionInformes	Datetime		Sí		Fecha de la recepción de los informes
<b>Fecha turno</b>	FechaTurno	Datetime		Sí		Fecha en la que se turna el asunto
<b>ID Secretario Auxiliar</b>	SecretarioAuxiliarID	Int (4)		Sí		Identificador del secretario al que le pertenece
<b>Fecha devolución oficialía</b>	FechaDevolucionOficialia	Varchar (100)		Sí		Fecha en la que la oficialía devolvió el asunto
<b>Fecha devolución informes</b>	FechaDevolucionMesasInformes	Datetime		Sí		Fecha en la que la mesa de informes devolvió el asunto
<b>ID tipo promoción</b>	TipoPromocionID	Int (4)		Sí		Identificador del tipo de promoción del asunto
<b>Nombre usuario</b>	UsuarioCreacion	Varchar (50)		No		Nombre del usuario que creó el asunto
<b>Nombre usuario actualización</b>	UsuarioActualizacion	Varchar (50)		Sí		Nombre del usuario que actualiza asunto
<b>Fecha creación asunto</b>	FechaCreacion	Datetime		No		Fecha de creación del asunto
<b>Fecha actualización asunto</b>	FechaActualizacion	Datetime		Sí		Fecha de última actualización del asunto

**Tabla 4.1.5.15 Promocion**

Tabla Acuerdo						
Nombre	Acrónimo	Tipo y Longitud	Llave	Nulos	Tablas Asociadas	Descripción
<b>ID Acuerdo</b>	AcuerdoID	Int (4)	PK	No		Identificador del acuerdo
<b>ID Asunto</b>	AsuntoID	Int(4)	FK	No	Asunto	Identificador del asunto
<b>Descripción acuerdo</b>	Acuerdo	Varchar (max)		Sí		Descripción del acuerdo
<b>ID Secretario Auxiliar</b>	SecretarioAuxiliarID	Int (4)		Sí		Identificador del secretario al que le pertenece
<b>Fecha acuerdo</b>	FechaAcuerdo	Datetime		Sí		Fecha de creación del acuerdo
<b>Observación</b>	Observaciones	Varchar (max)		Sí		Observaciones sobre el acuerdo
<b>Nombre del dueño</b>	PertenencialID	Int (4)		Sí		Identificador de persona a la que pertenece
<b>Documento</b>	Documento	Varbinary (max)		Sí		Documento digital del acuerdo
<b>Bit firma</b>	Firmado	Bit (1)		Sí		Define si está firmado o no
<b>Fecha firma</b>	FechaFirma	Datetime		Sí		Fecha de firma del acuerdo
<b>Bit devolución</b>	Devuelto	Bit (1)		Sí		Define si se ha devuelto el asunto o no
<b>Fecha devolución</b>	FechaDevuelto	Datetime		Sí		Fecha de devolución del asunto
<b>Nombre usuario</b>	UsuarioCreacion	Varchar (50)		No		Nombre del usuario que creó el asunto
<b>Nombre usuario</b>	UsuarioActualizacion	Varchar (50)		Sí		Nombre del usuario que actualiza

<b>actualización</b>						asunto
<b>Fecha creación asunto</b>	FechaCreacion	Datetime		No		Fecha de creación del asunto
<b>Fecha actualización asunto</b>	FechaActualizacion	Datetime		Sí		Fecha de última actualización del asunto

**Tabla 4.1.5.16 Acuerdo**

<b>Tabla AcuerdoTipoAcuerdo</b>						
<b>Nombre</b>	<b>Acrónimo</b>	<b>Tipo y Longitud</b>	<b>Llave</b>	<b>Nulos</b>	<b>Tablas Asociadas</b>	<b>Descripción</b>
<b>ID Acuerdo tipo</b>	AcuerdoID	Int (4)	PK	No		Identificador del acuerdo tipo de acuerdo
<b>ID tipo de acuerdo</b>	TipoAcuerdoID	Int (4)	FK	No	TipoAcuerdo	Identificador del tipo de acuerdo

**Tabla 4.1.5.17 AcuerdoTipoAcuerdo**

<b>Tabla AcuerdoPromocion</b>						
<b>Nombre</b>	<b>Acrónimo</b>	<b>Tipo y Longitud</b>	<b>Llave</b>	<b>Nulos</b>	<b>Tablas Asociadas</b>	<b>Descripción</b>
<b>ID Acuerdo Promoción</b>	PromocionID	Int (4)	PK	No		Identificador del Acuerdo de la promoción
<b>ID Acuerdo</b>	AcuerdoID	Int (4)	FK	No	Acuerdo	Identificador del acuerdo

**Tabla 4.1.5.18 AcuedoPromocion**

<b>Tabla AcuerdoTema</b>						
<b>Nombre</b>	<b>Acrónimo</b>	<b>Tipo y Longitud</b>	<b>Llave</b>	<b>Nulos</b>	<b>Tablas Asociadas</b>	<b>Descripción</b>
<b>ID Acuerdo</b>	AcuerdoID	Int (4)	FK	No	Acuerdo	Identificador del acuerdo
<b>ID Tema</b>	TemaID	Int (4)	FK	No	Tema	Identificador del tema

**Tabla 4.1.5.19 AcuerdoTema**

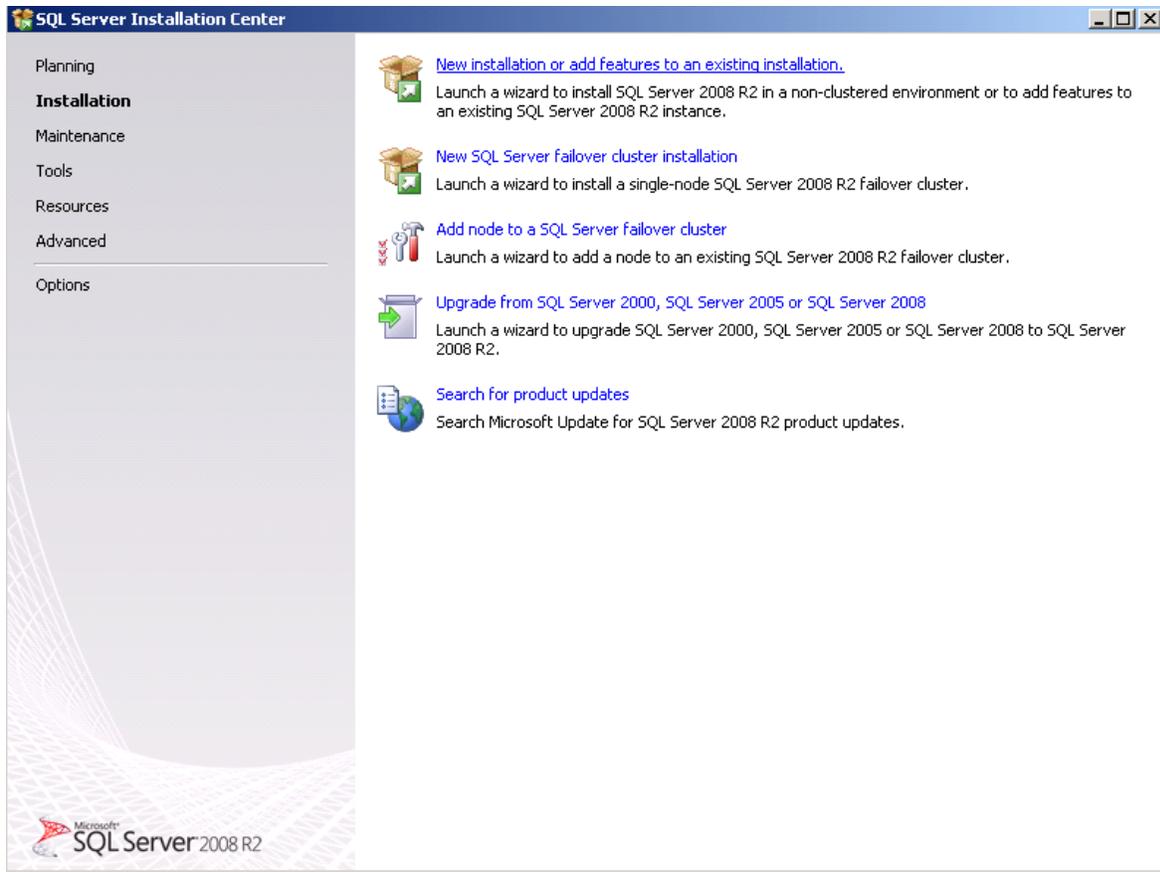
## **4.2 Implementación de Base de Datos**

### **Instalación SQL Server 2008 R2**

El primer paso para la instalación del manejador es contar con el siguiente software:

- SQL Server 2008 R2 Enterprise Edition x64 RTM

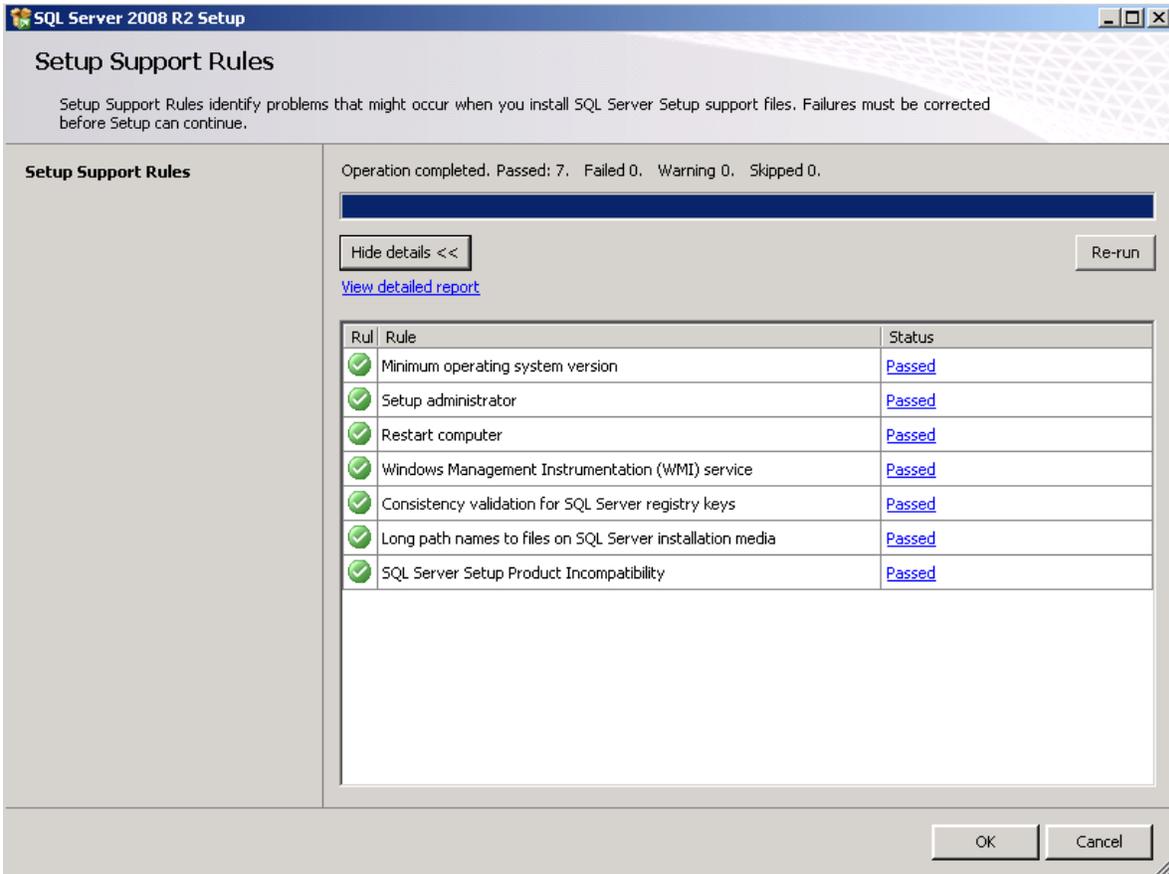
Se debe correr el setup que viene en el paquete para comenzar la instalación. Aparecerá una pantalla que nos guiará durante el proceso de instalación, la cual se ve representada en la figura 4.2.1.



**Figura 4.2.1** Pantalla de inicio de instalación de SQL Server 2008 R2

Una vez que nos encontramos en este paso, debemos elegir la opción “Nueva Instalación” ó “New Installation or add features to an existing installation” con la que instalaremos una nueva instancia de SQL Server 2008 R2 a nuestro equipo.

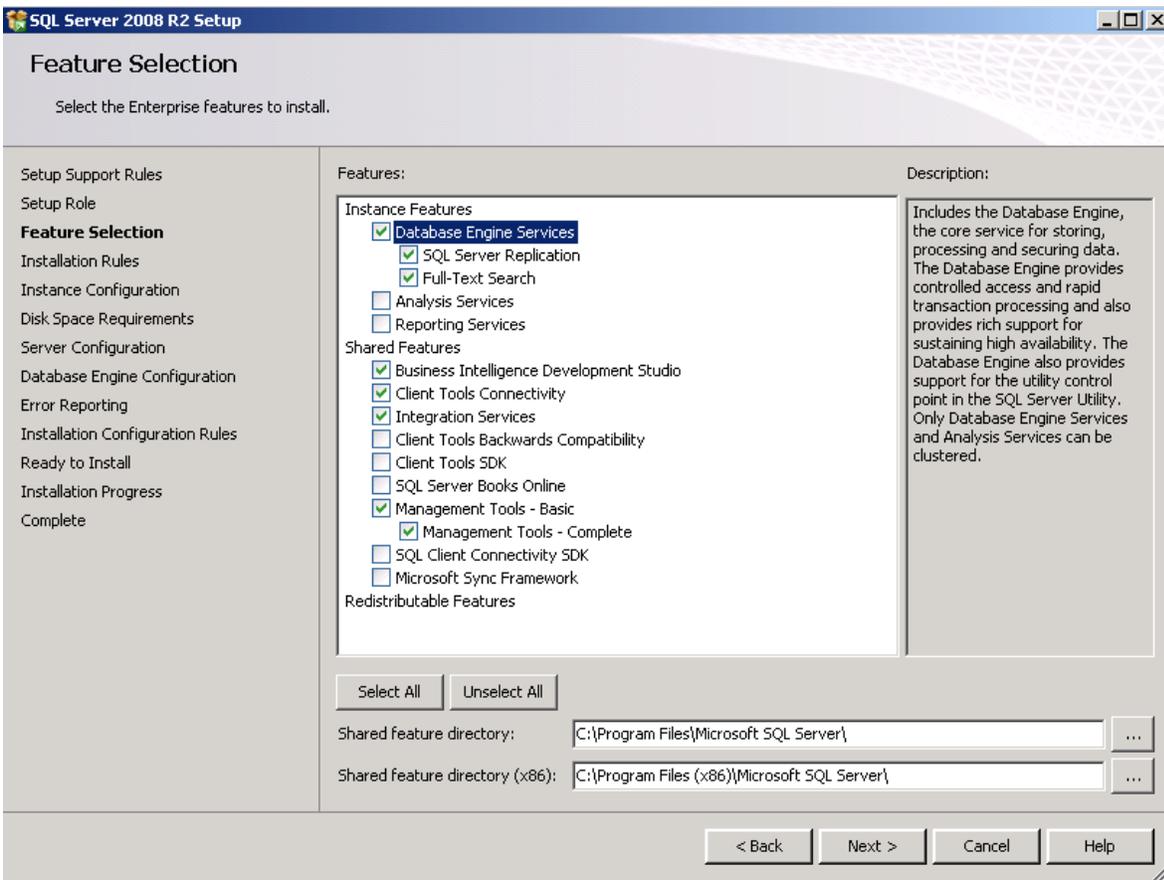
Después de elegir esta opción, el programa procederá a hacer las verificaciones que son necesarias para proceder a la instalación. Verifica que el sistema operativo sea compatible, que los requerimientos mínimos de hardware y software se cumplan, entre otras características que se deben de cumplir antes de que empiece la instalación del manejador. Si todo se encuentra en orden y el software verificó que se cumplan con todos los requisitos previos a la instalación, aparecerá una nueva pantalla donde nos dirá que se puede proseguir con el siguiente paso de la instalación, de no ser así, nos dirá cuáles son los requerimientos que nos hacen falta para la instalación. La figura 4.2.1 muestra esta pantalla.



**Figura 4.2.2 Verificación de requerimientos de instalación**

Después de estos pasos de verificación, lo siguiente es validar el producto. Para esto, el software nos pide una licencia para validar que el producto sea original y que se tenga una licencia para hacer uso del mismo.

Después de la validación, se deben elegir las características que queremos que se instalen en nuestra instancia de SQL Server. La figura 4.2.3 muestra las características que podemos elegir para su instalación.



**Figura 4.2.3 Selección de características SQL Server**

## Database Engine Services

Se trata del motor relacional de SQL Server, justo lo que queremos instalar. Como Buena práctica, siempre es recomendable en cualquier instalación de cualquier software, solo instalar aquellas características de las cuales se va a hacer uso.

## Business Intelligence Development Studio

También conocido como BIDS, se trata de una versión de Visual Studio con soporte para la creación de proyectos de inteligencia de negocio como Integration Services, Reporting Services o Analysis Services.

## **Client Tools Connectivity**

Librerías de acceso SQLNCLI para comunicación entre clientes y servidores SQL Server

## **Integration Services**

Pese a que este servicio no es puramente relacional, siempre es útil instalarlo junto al motor relacional porque es habitual realizar procesos de carga y transformación de datos utilizando SQL Server Integration Services.

## **Management Tools – Basic**

Para el caso de nuestro sistema, es buena idea tener instaladas las herramientas de administración de SQL Server para emergencias (por si solo pudiéramos logear localmente por cualquier motivo). Se instalarán: SQL Server Management Studio, SQLCMD y el proveedor powershell para SQL Server (sqlps.exe).

## **Management Tools – Complete**

Añade soporte extra para SQL Server Management Studio con conexión a Reporting Services, Analysis Services e Integration Services.

El siguiente paso en la instalación es el de configurar nuestra instancia del manejador. Los parámetros que son necesarios de configurar son:

- Nombre de la instancia a instalar.
- ID de la instancia.
- Directorio donde se instalará la aplicación SQL Server.
- Configuración de los servicios que correrán al momento de lanzar la aplicación.

- Idioma tanto del programa como el de default para las Bases de Datos.
- Método de autenticación (aquí hay 3 formas: mediante la autenticación automática al iniciar la sesión de Windows, mediante un usuario y contraseña y la tercera es mediante una mezcla entre las primeras 2).
- La ruta de los directorios donde se instalarán las carpetas necesarios para archivar la información que se vaya generando (log, temporales, data, etc.).
- Habilitar el tipo de dato FILESTREAM.

En las figuras 4.2.4, 4.2.5, 4.2.6 y 4.2.7 podemos observar las pantallas donde se configuran las características antes mencionadas para nuestra instancia de SQL Server.

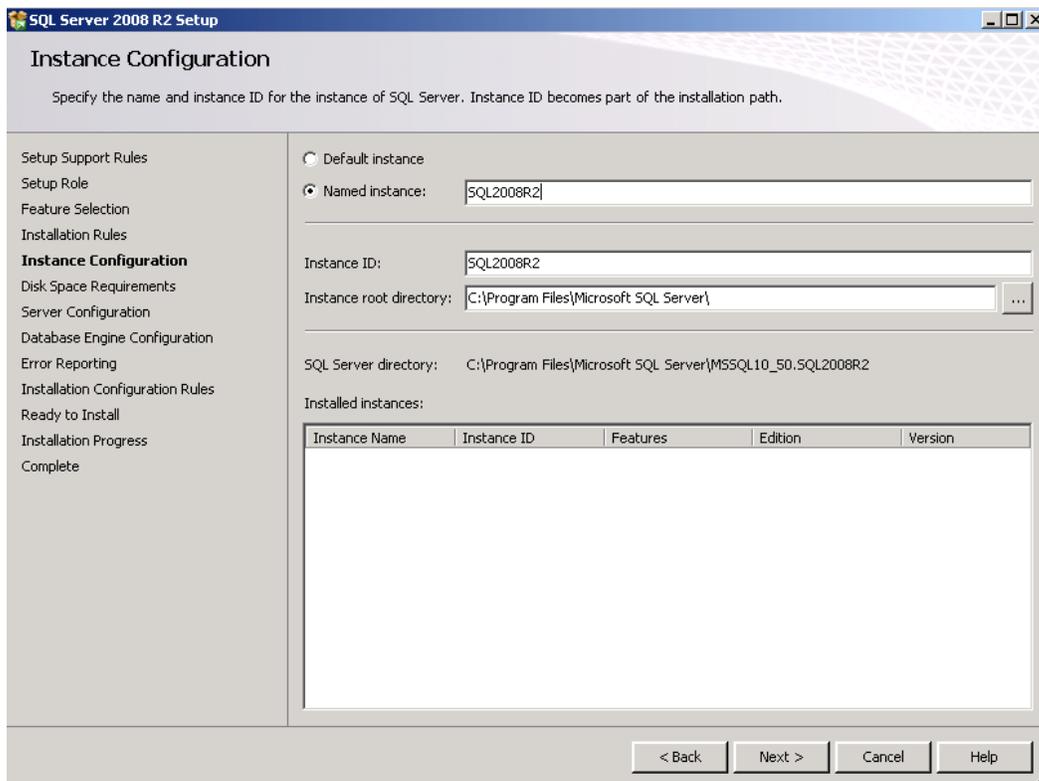


Figura 4.2.4

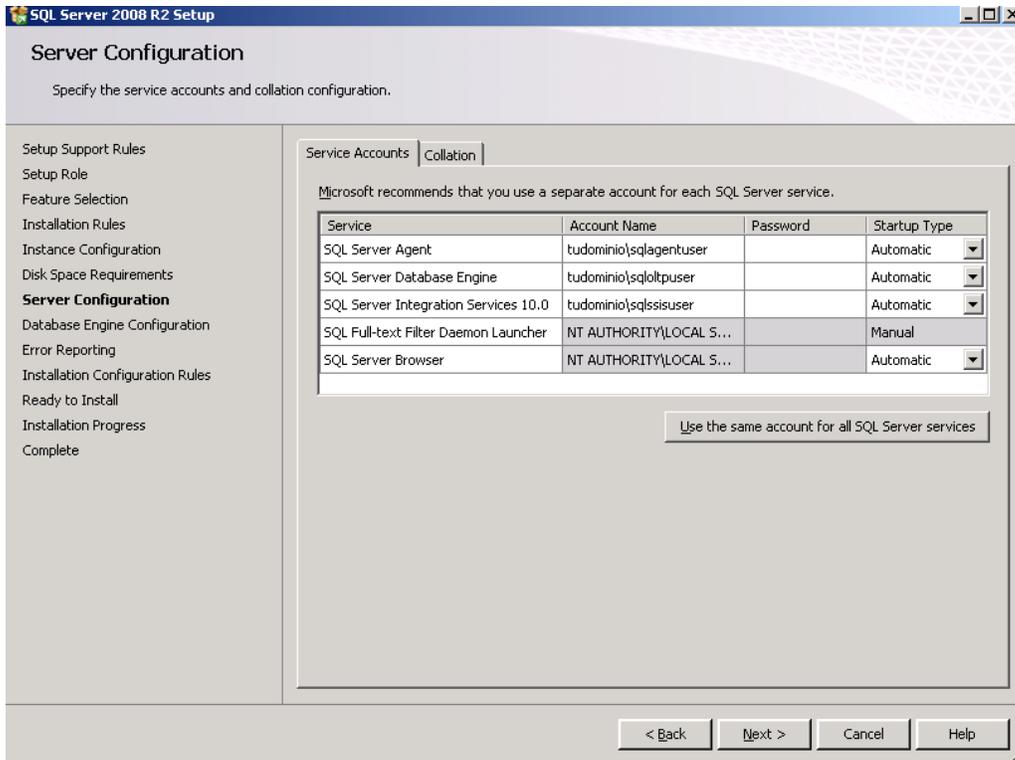


Figura 4.2.5

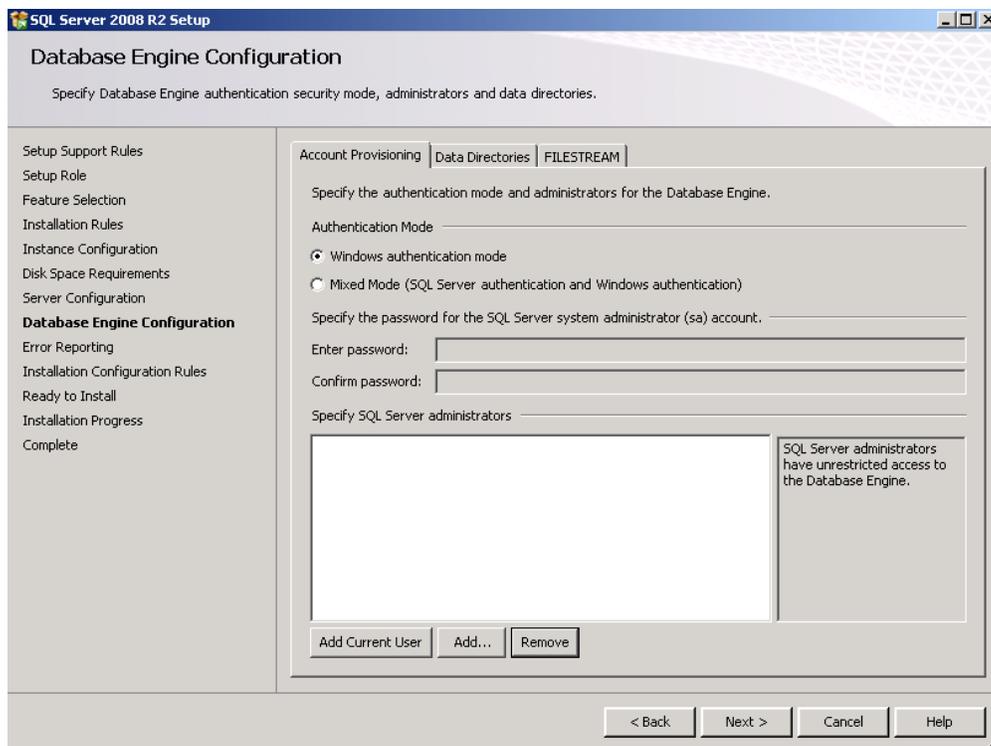
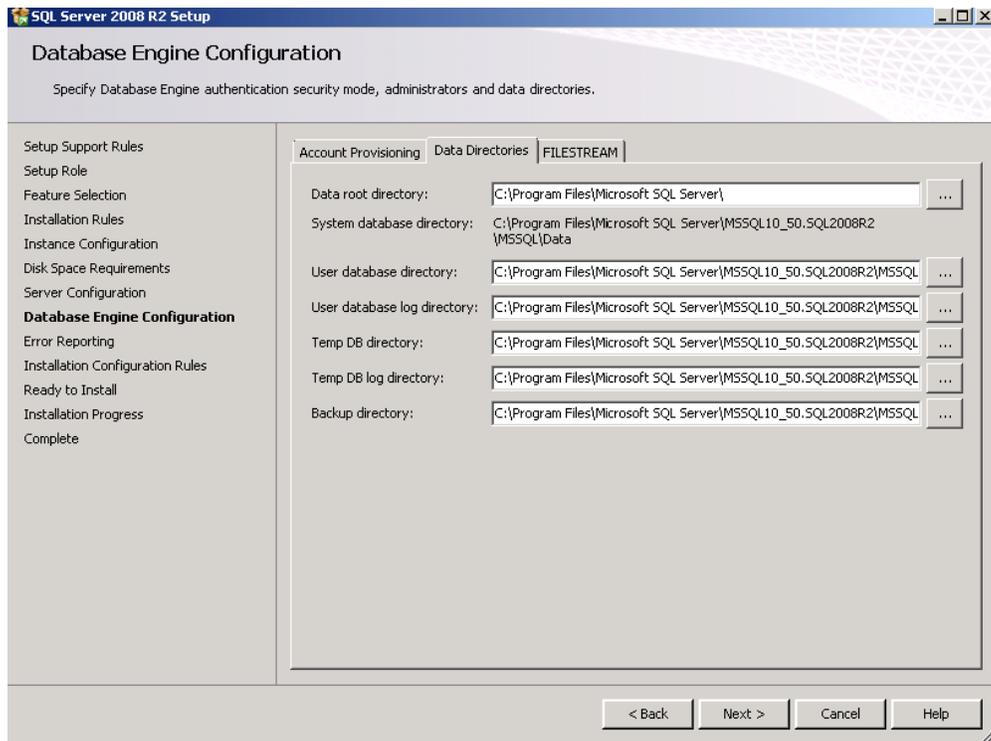


Figura 4.2.6



**Figura 4.2.7**

Una vez configurados todos los parámetros de nuestra instancia, el instalador verificará estas configuraciones y si éstas son válidas, continuará con el proceso de instalación. Instalará los archivos, creará las carpetas necesarias, y activará las configuraciones para que nuestro manejador pueda ser utilizado de la manera en el que lo configuramos. La figura 4.2.8 muestra la pantalla que aparecerá si la instalación ha finalizado con éxito. De ser así, no será necesario reiniciar y si se requiere, es posible inicializar SQL Server 2008 R2 para utilizarlo en ese momento.

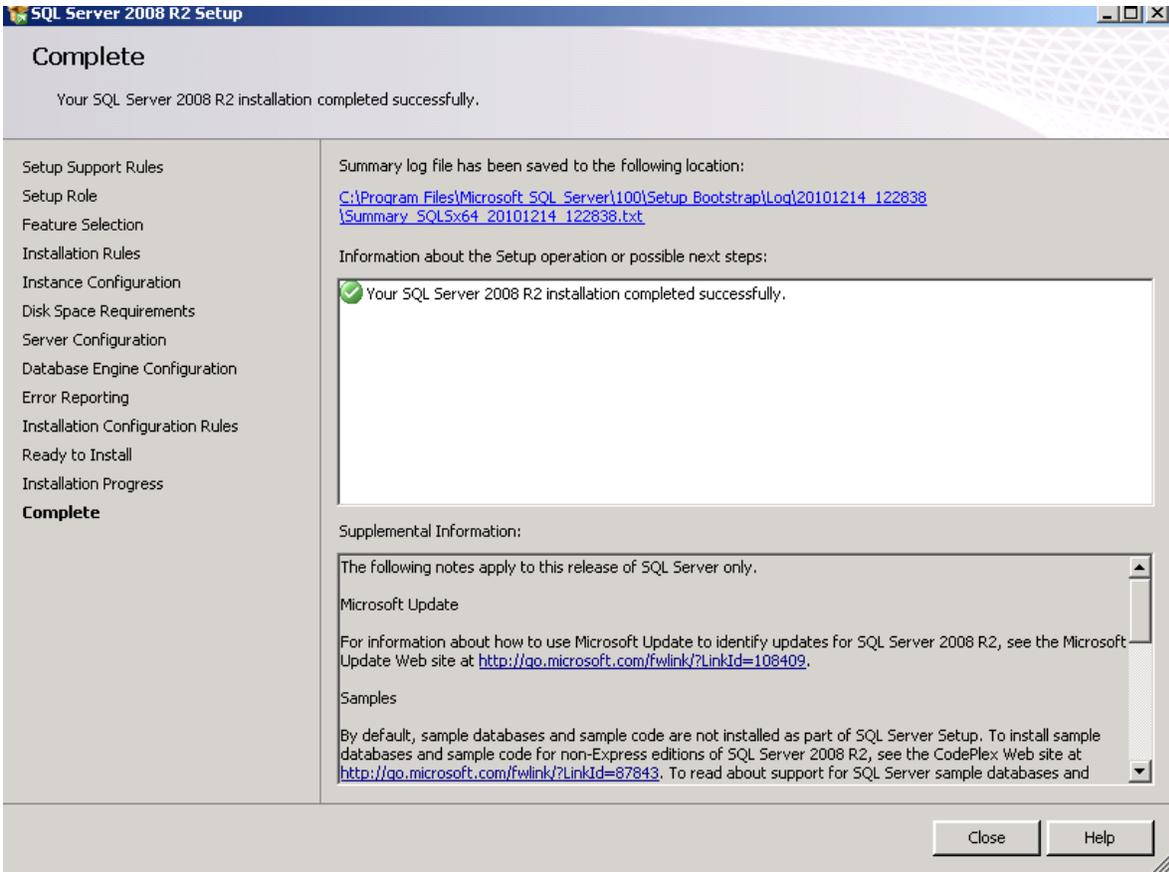


Figura 4.2.8

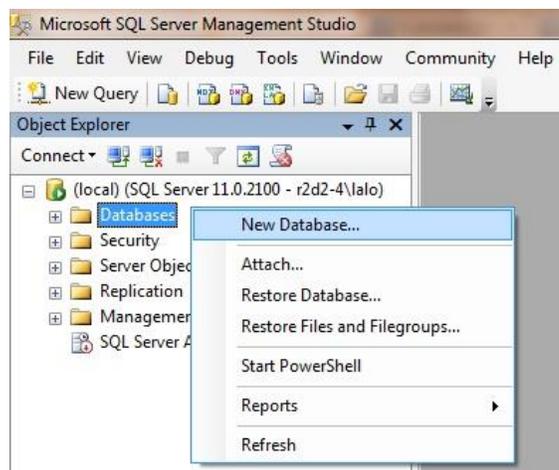
## CREACIÓN DE BASE DE DATOS

El siguiente paso después de la instalación de nuestro manejador, es la creación de la base de datos que utilizaremos para nuestro sistema. Para ello, abriremos nuestra aplicación “Microsoft SQL Server Management Studio” que se instaló dentro de la carpeta de SQL Server 2008 R2, como se muestra en la figura 4.2.9.



**Figura 4.2.9**

Después de iniciar la aplicación y de introducir nuestro usuario y contraseña para validarnos dentro del manejador, del lado izquierdo aparecerá la ventana de “Explorador de objetos”. Ahí, debemos de colocar el mouse sobre la carpeta “Databases” y dar un clic derecho sobre la carpeta. Aparecerá una nueva ventana como la que se muestra en la figura 4.2.10, donde debemos escoger la opción “New Database”.



**Figura 4.2.10**

Al elegir esta opción, aparecerá una nueva ventana (Figura 4.2.11), donde escribiremos el nombre de la nueva base de datos y elegiremos los parámetros que utilizaremos para la base.

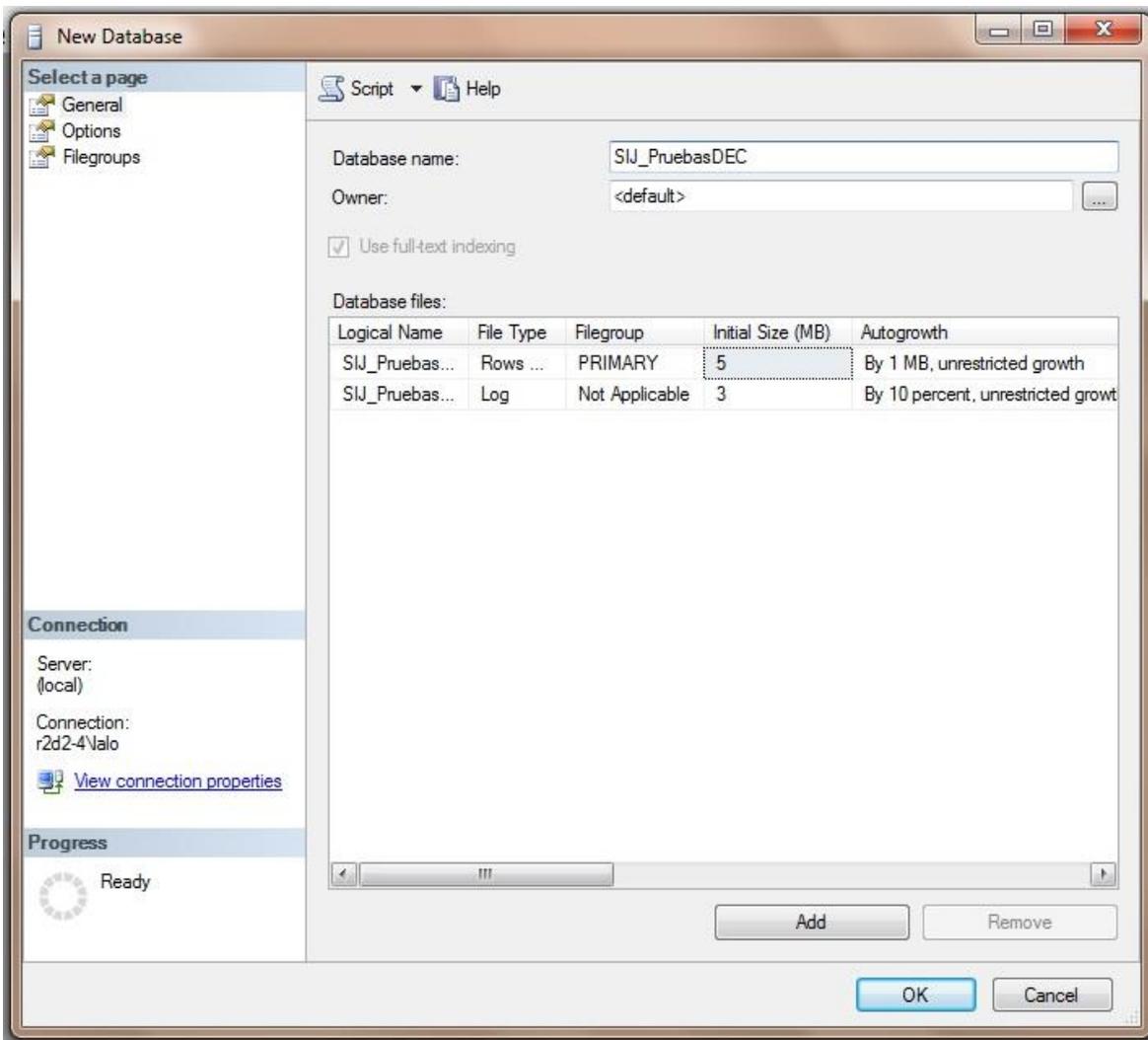


Figura 4.2.11 Creación de una base de datos.

## CREACIÓN DE UNA TABLA

Una tabla se puede crear de dos maneras mediante código y utilizando el asistente, en las imágenes siguientes se muestra esta última.

En la primer imagen (figura 4.2.12), se define el nombre de la tabla y algunas de sus propiedades, en la siguiente (figura 4.2.13), se definen los campos y los tipos de datos que tiene cada uno.

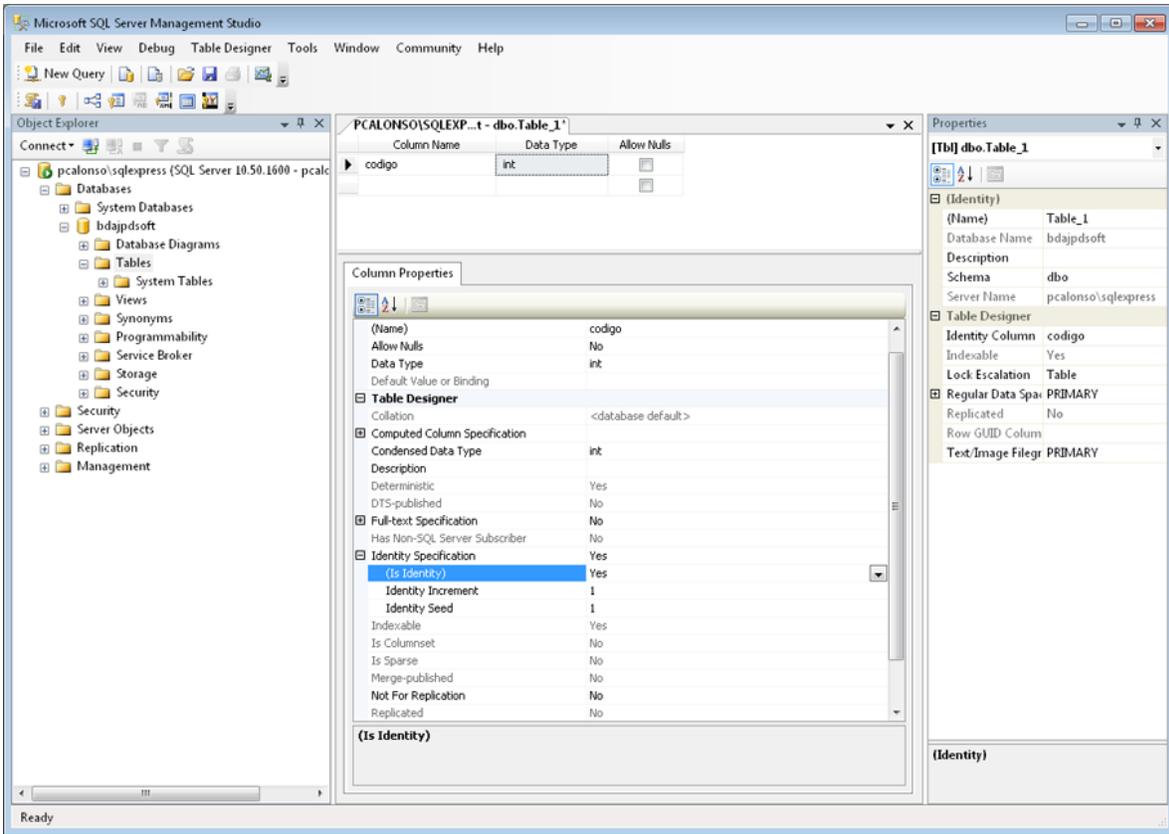
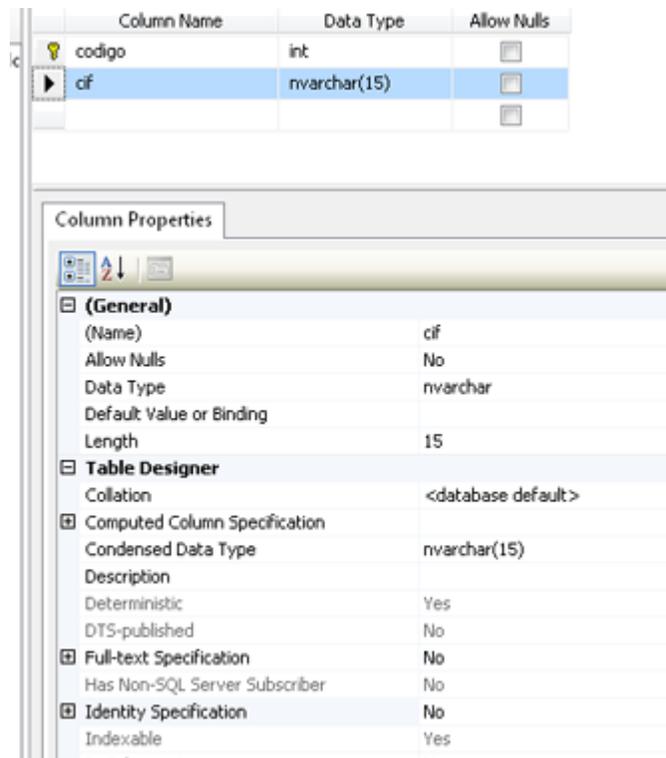


Figura 4.2.12

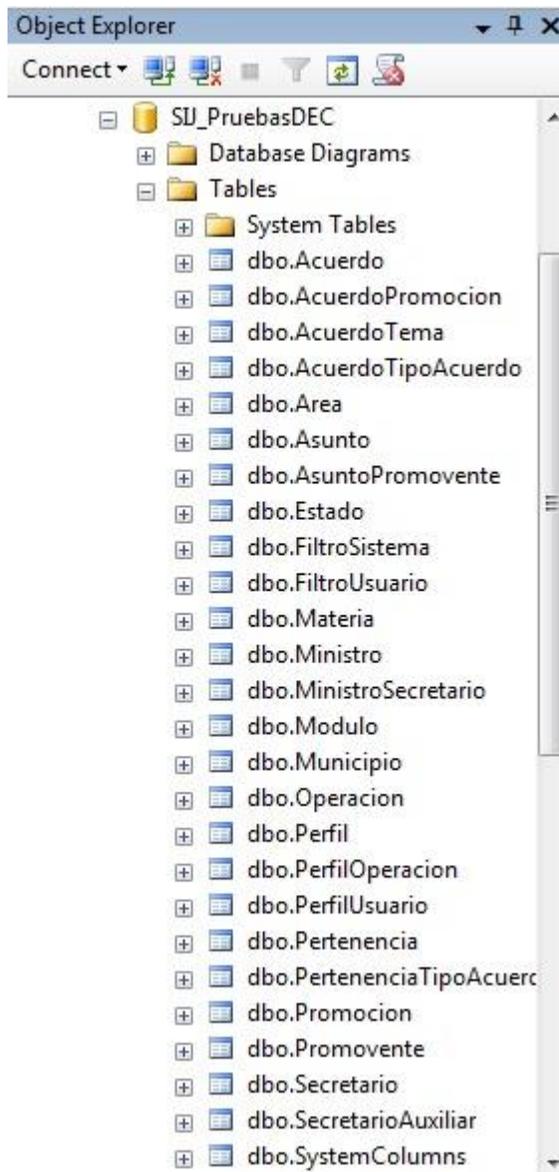


**Figura 4.2.13**

### Mediante Código SQL:

```
CREATE TABLE "nombre_tabla"
("columna 1" "tipo_de_datos_para_columna_1",
"columna 2" "tipo_de_datos_para_columna_2",
... )
```

Una vez creadas las tablas, se pueden consultar en el árbol de tablas que se encuentra dentro de la ventana del “Explorador de Objetos” como se ve en la figura 4.2.14.



**Figura 4.2.14**

## **Creación de Funciones**

Una función es un bloque de código PL/SQL que tiene las mismas características que un procedimiento almacenado. La diferencia estriba que una función devuelve un valor al retornar. Al devolver un valor puede ser llamada como parte de una expresión.

## Sintaxis

```
CREATE FUNCTION [owner_name.] function_name  
( [{ @parameter_name}scalar_parameter_type [= default]} [...n]])  
RETURNS scalar_return_type  
[WITH <function_option>::={SCHEMABINDING | ENCRYPTION}]  
[AS]  
BEGIN  
function_body  
RETURN scalar_expression  
END
```

## Árbol de Funciones

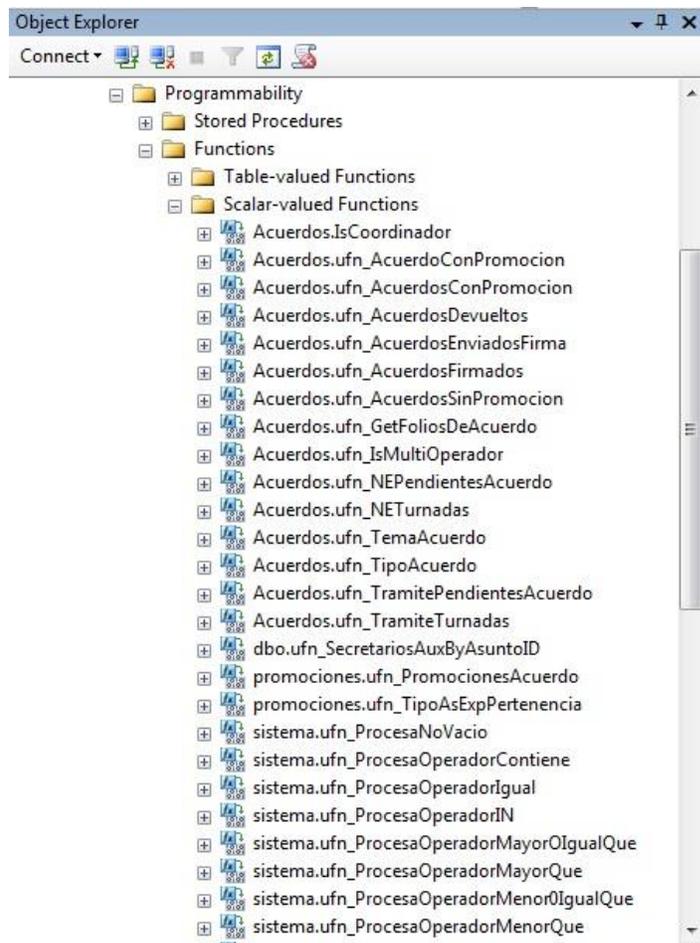


Figura 4.2.15 Árbol de funciones

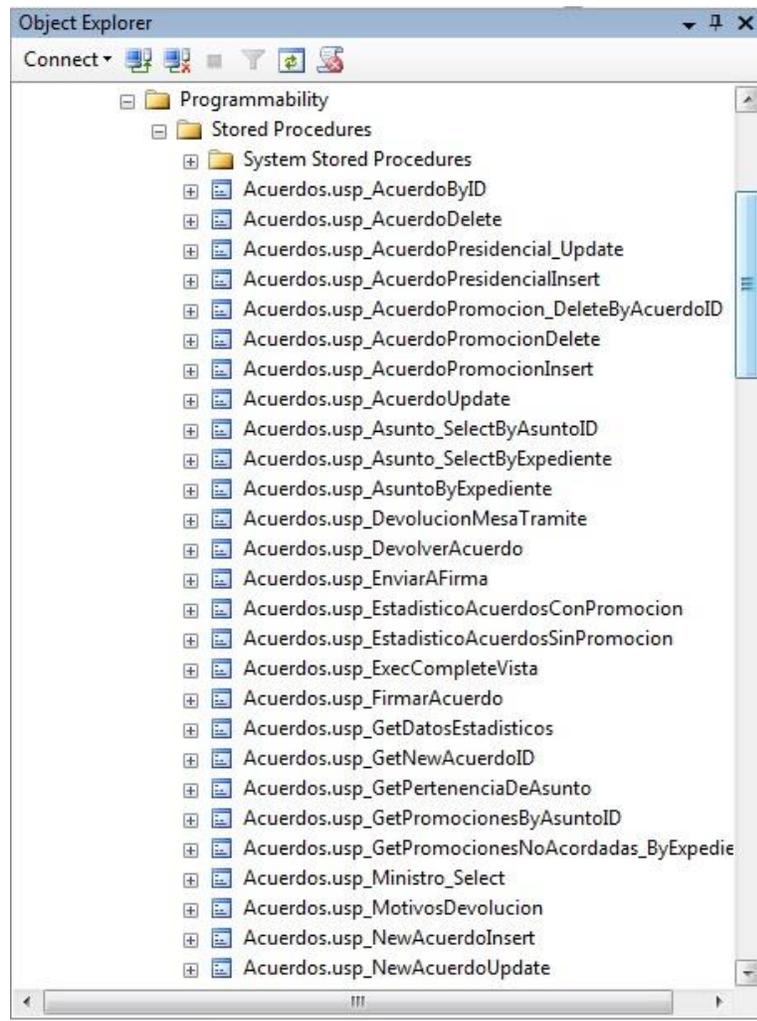
## Creación de Procedimientos Almacenados

Un procedimiento almacenado es un objeto perteneciente a una base de datos, que contiene un conjunto de instrucciones SQL, tanto de consulta, como de manipulación de datos, como de control de la secuencia del programa, asociados a un nombre, y que son ejecutados en conjunto. Puede contener parámetros tanto de entrada como de salida (parámetros pasados por referencia), así como devolver un valor de retorno.

### Sintaxis

```
CREATE PROC [ EDURE ] [ owner. ] procedure_name[ ; number ]
[ { @parameter data_type}
[ VARYING ] [ = default ] [ OUTPUT ] ] [ ,...n ]
[ WITH
{ RECOMPILE | ENCRYPTION | RECOMPILE , ENCRYPTION } ]
[ FOR REPLICATION ]
AS sql_statement [ ...n ]
```

## Árbol de Procedimientos Almacenados



**Figura 4.2.16** Árbol de Procedimientos Almacenados

### Creación de Vistas

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto. Almacena una consulta como un objeto para utilizarse posteriormente.

## Sintaxis

```
CREATE VIEW "NOMBRE_VISTA" AS "Instrucción SQL"
```

## Árbol de Vistas

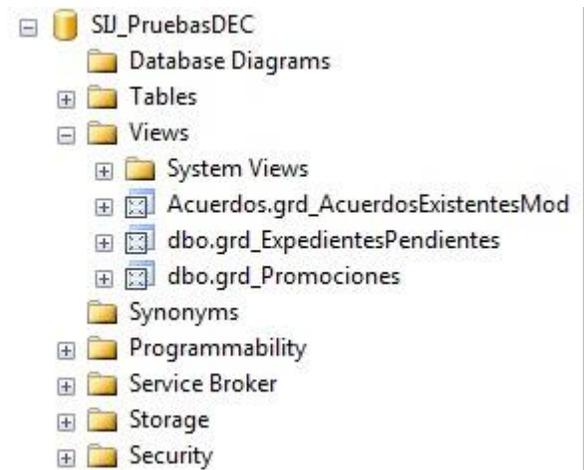


Figura 4.2.17 Árbol de Vistas

## 4.3 El diseño de la interfaz del usuario

En el diseño de nuestro sistema, la interfaz de usuario es muy importante, ya que los usuarios que utilizarán el sistema no son personas que tenga un amplio conocimiento en cuestión de aplicaciones, paquetería o sistemas informáticos.

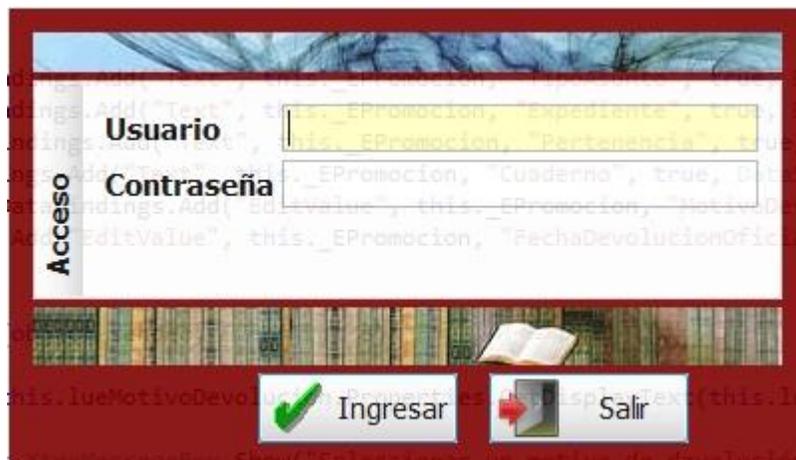
Debido a esto, es importante que la interfaz gráfica de nuestro sistema sea simple, práctica y muy intuitiva para que sea accesible a cualquier persona que deba manejarla. A continuación se dará una explicación de las pantallas que componen nuestro sistema, de acuerdo al módulo que corresponde. Como ya hemos explicado, nuestro sistema se divide en 3 módulos:

- Módulo “Ingreso de Asuntos”

- Módulo “Promociones”
- Módulo “Acuerdos”

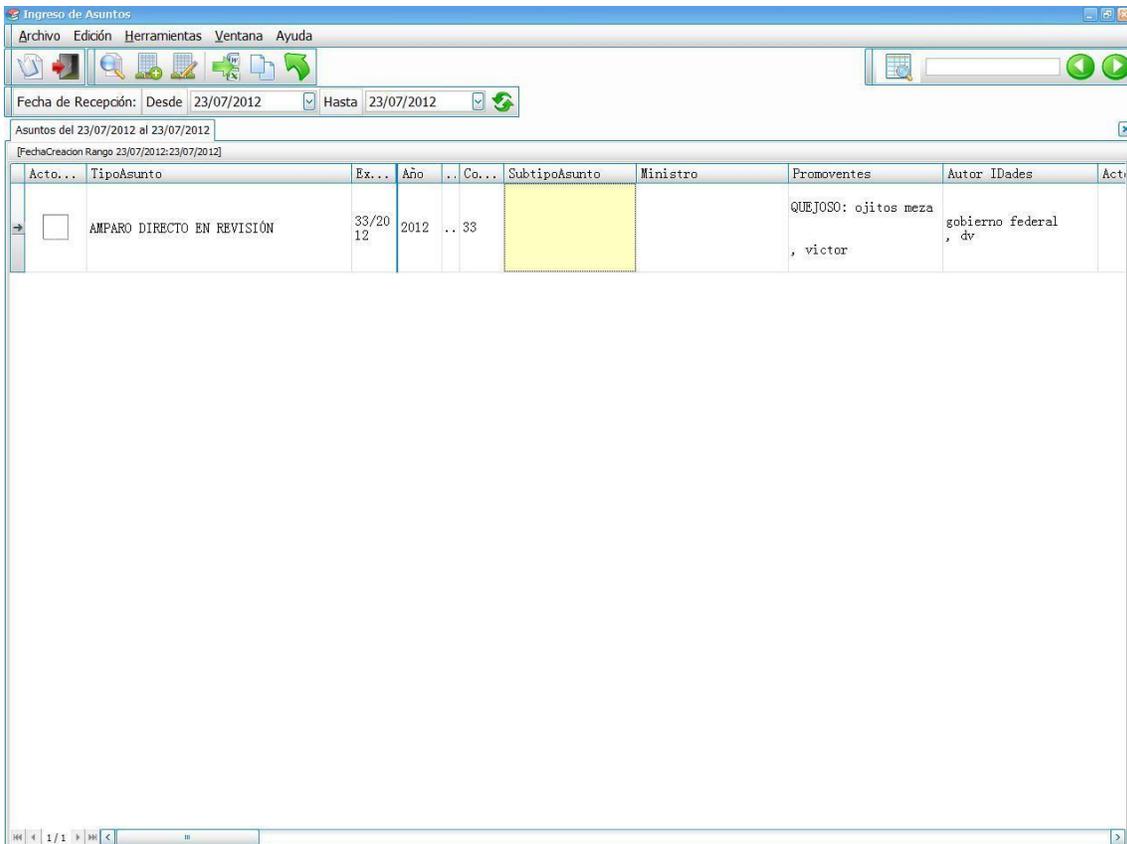
### **Módulo “Ingreso de Asuntos”**

La primera pantalla que aparecerá en nuestro sistema, en el módulo de Ingreso de Asuntos será la del Login. Aquí, el usuario introducirá un nombre de usuario y una contraseña para acceder al sistema. También hay un botón que nos permite salir de la aplicación en caso de que se desee cerrar la aplicación. En la figura 4.3.1 se muestra ésta primera pantalla.



**Figura 4.3.1 Pantalla de acceso al sistema**

Una vez validado el usuario y la contraseña, a continuación se presentará La pantalla principal del módulo donde se desplegará el menú de acciones y el área de trabajo. En la figura 4.3.2 se muestra esta pantalla.



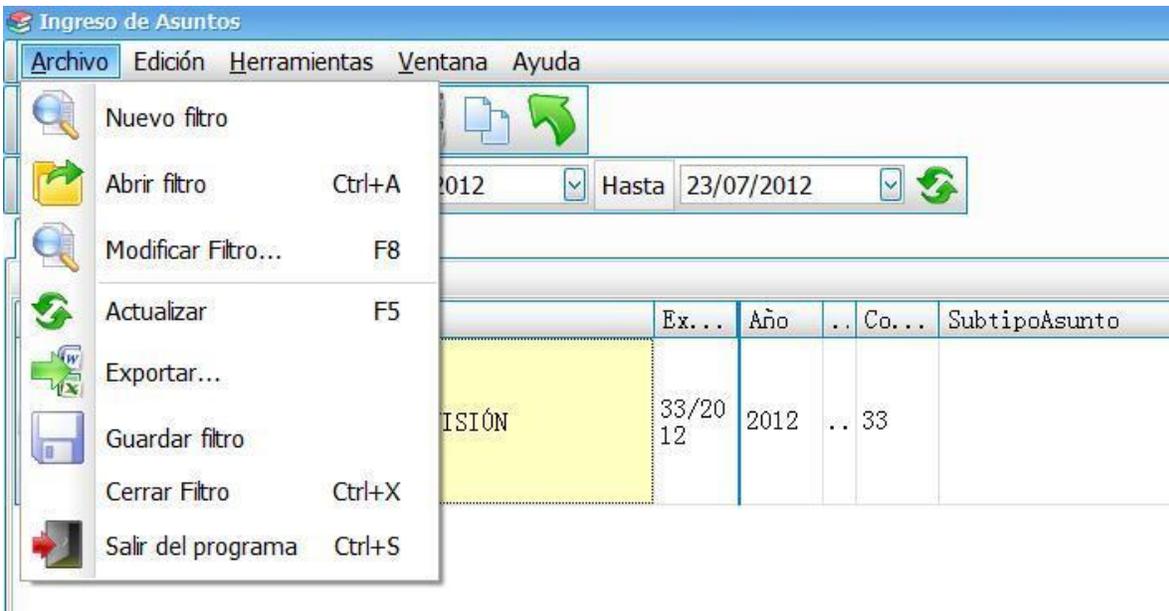
**Figura 4.3.2 Ventana Principal del módulo “Ingreso de Asuntos”**

En la pantalla principal se encuentran todas las opciones que existen en éste módulo. En primera instancia tenemos el menú principal en el cual se encuentran los siguientes elementos:

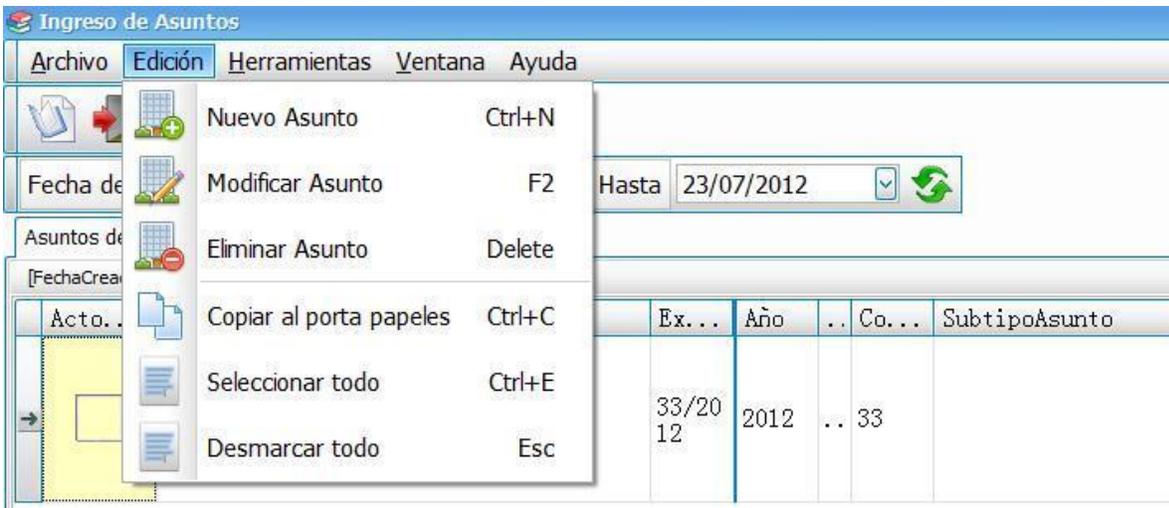
- **Archivo:** Aquí se encuentran los botones para abrir un archivo, para hacer una búsqueda o filtro, para modificar o eliminar dicho filtro, para actualizar el área de trabajo y para salir de la aplicación.
- **Edición:** En este elemento encontramos las opciones para hacer crear, modificar o eliminar un asunto, así como también las opciones de selección, copia al portapapeles y la de marcar o desmarcar casillas.
- **Herramientas:** Aquí encontramos las opciones de configuración del programa, búsquedas de archivos o expedientes y organización de los registros en el área de trabajo.

- **Ventana:** Debido a que la aplicación puede tener varias instancias corriendo simultáneamente en la misma máquina, aquí se da la opción para organizar las instancias de una forma que sea útil al usuario.
- **Ayuda:** Como en todas las pantallas principales de nuestro sistema, existe éste botón para brindarle ayuda o una explicación de los componentes del sistema al usuario.

En las figuras 4.3.3 y 4.3.4 se muestran los elementos desplegados del menú Archivo y del menú Edición respectivamente.



**Figura 4.3.3 Menú Archivo desplegado.**



#### 4.3.4 Menú Edición desplegado.

Algunos elementos de este menú, al elegirlos, desplegarán nuevas ventanas donde se seguirá el proceso para finalizar la acción que se necesite. Por ejemplo, en la opción de “Nuevo Asunto”, se abrirá una nueva ventana donde podremos dar de alta un expediente o asunto con las características que se requieran o las que se puedan dar en ese momento. Es el primer paso para crear un nuevo proceso en el flujo de información en nuestro sistema. La figura 4.3.5 muestra la ventana que aparecerá al elegir esta opción.

Alta de expediente

\* Tipo de Asunto: <Elija una opción>

Subtipo de Asunto: <Elija una opción>

\* Expediente: / 2012 - Ext. \* Fecha y hora del reloj fechador: 23/07/2012 9:57

Órgano de radicación: <Elija una opción>

Ubicación física (Destino): PRIMERA SALA

\* Número de registro: Número de Expediente Único Nacional

Número de Oficio o Escrito: Estado de la federación: <Elija una opción> Municipio: [Vacío]

Materia y SubMateria:

- ADMINISTRATIVA
  - AGRARIA
  - FISCAL
  - ELECTORAL
  - TRATADOS INTERNACIONALES
  - ASCASCAS
  - CIVIL
    - ELECTORAL
    - FAMILIAR
    - TRATADOS INTERNACIONALES
  - COMÚN
    - ELECTORAL
    - TRATADOS INTERNACIONALES
  - CONSTITUCIONAL
    - ELECTORAL
    - TRATADOS INTERNACIONALES
  - DE TRABAJO
    - ELECTORAL
    - TRATADOS INTERNACIONALES
  - PENAL
    - ELECTORAL
    - TRATADOS INTERNACIONALES

Cuadernos y Anexos

Acto reclamado

Garantías violadas ...

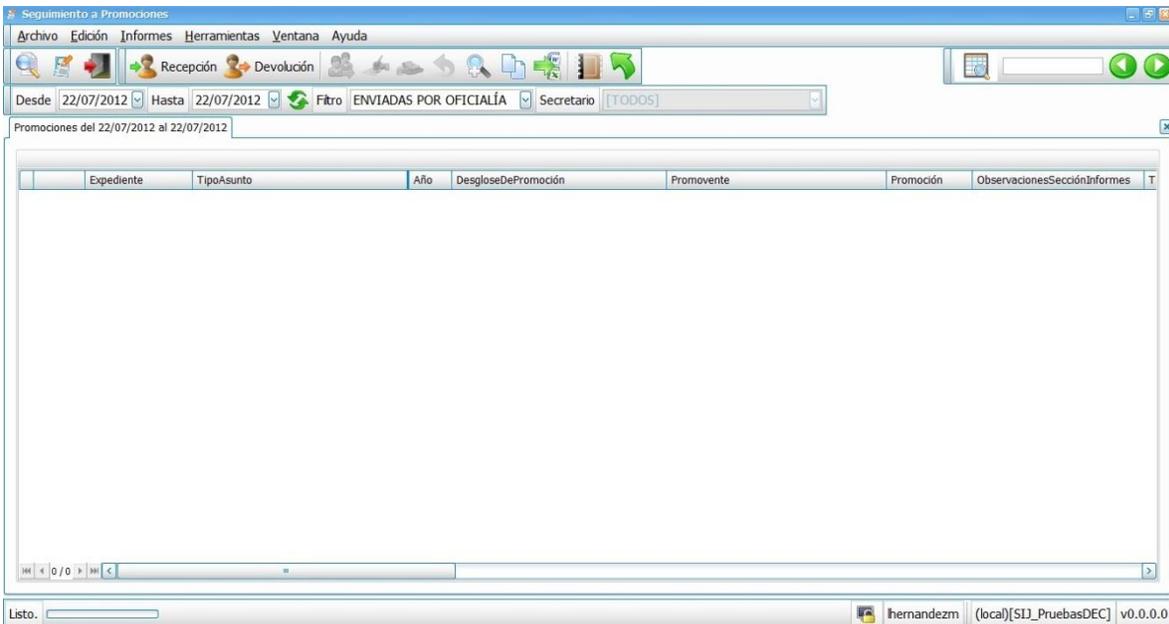
Derechos humanos relacionados ...

Aceptar Salir

Figura 4.3.5 Sub-ventana de “Alta de expedientes”

## Módulo de Promociones

Para éste módulo, nuevamente la pantalla inicial será la del Login. Una vez que el usuario se haya validado dentro del sistema, aparecerá la ventana principal del módulo. Aquí estará el menú principal y el área de trabajo. En la figura 4.3.6 se muestra esta pantalla.



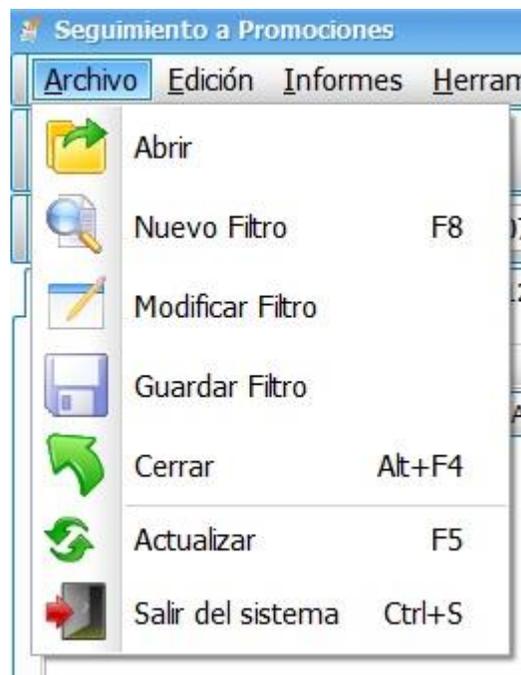
**Figura 4.3.6 Pantalla principal del módulo Promociones**

En la pantalla principal se encuentran todas las opciones que existen en éste módulo. En primera instancia tenemos el menú principal en el cual se encuentran los siguientes elementos:

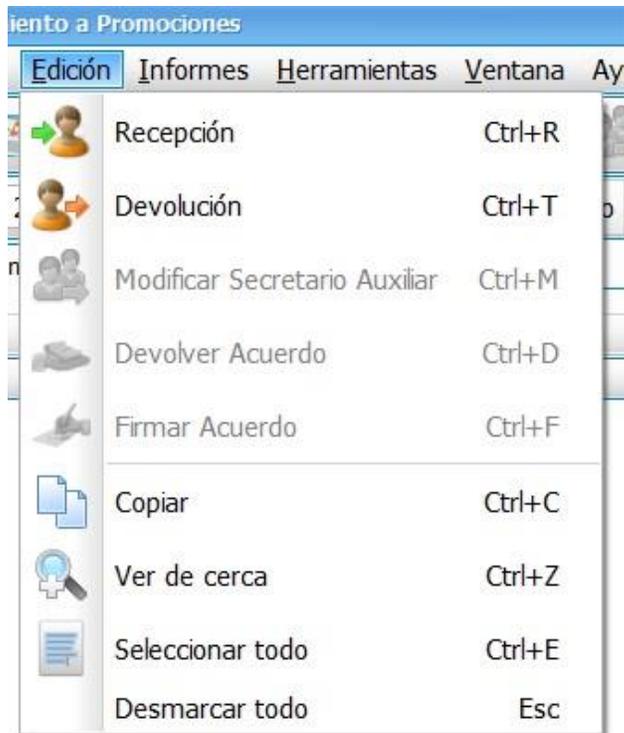
- **Archivo:** Aquí se encuentran los botones para abrir un archivo, para hacer una búsqueda o filtro, para modificar o eliminar dicho filtro, para actualizar el área de trabajo y para salir de la aplicación.
- **Edición:** En éste elemento encontramos las opciones para hacer la recepción o la devolución de los documentos ingresados previamente. También se encuentra la opción de copiar texto, hacer un zoom, seleccionar o desmarcar texto.
- **Informes:** Sólo existe una opción en éste elemento que es la de dar la ficha del expediente. Es decir, nos muestra los datos generales del asunto que tengamos marcado para su revisión.
- **Herramientas:** Aquí encontramos las opciones de configuración del programa, búsquedas de archivos o expedientes y organización de los registros en el área de trabajo.

- **Ventana:** Debido a que la aplicación puede tener varias instancias corriendo simultáneamente en la misma máquina, aquí se da la opción para organizar las instancias de una forma que sea útil al usuario.
- **Ayuda:** Como en todas las pantallas principales de nuestro sistema, existe éste botón para brindarle ayuda o una explicación de los componentes del sistema al usuario.

En las figuras 4.3.7 y 4.3.8 se muestran los elementos desplegados del menú Archivo y del menú Edición.

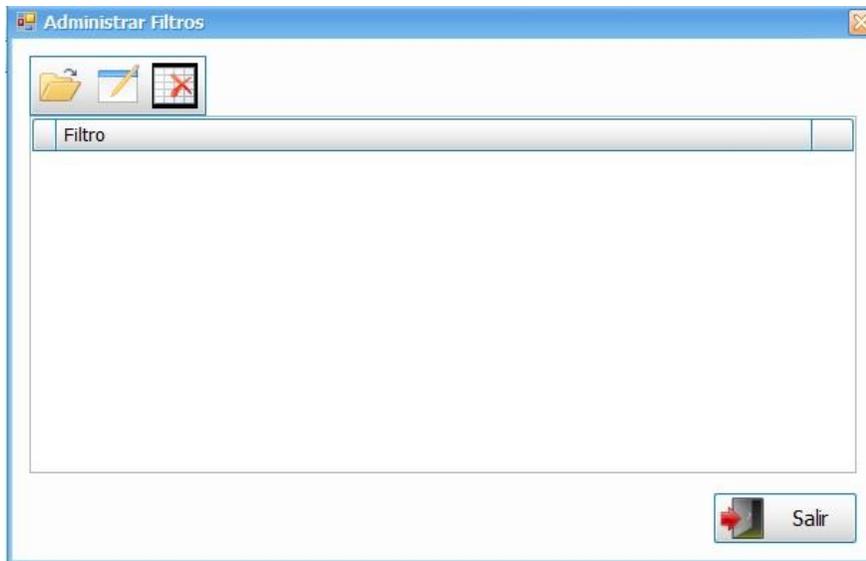


**Figura 4.3.7 Menú Archivo con elementos desplegados**



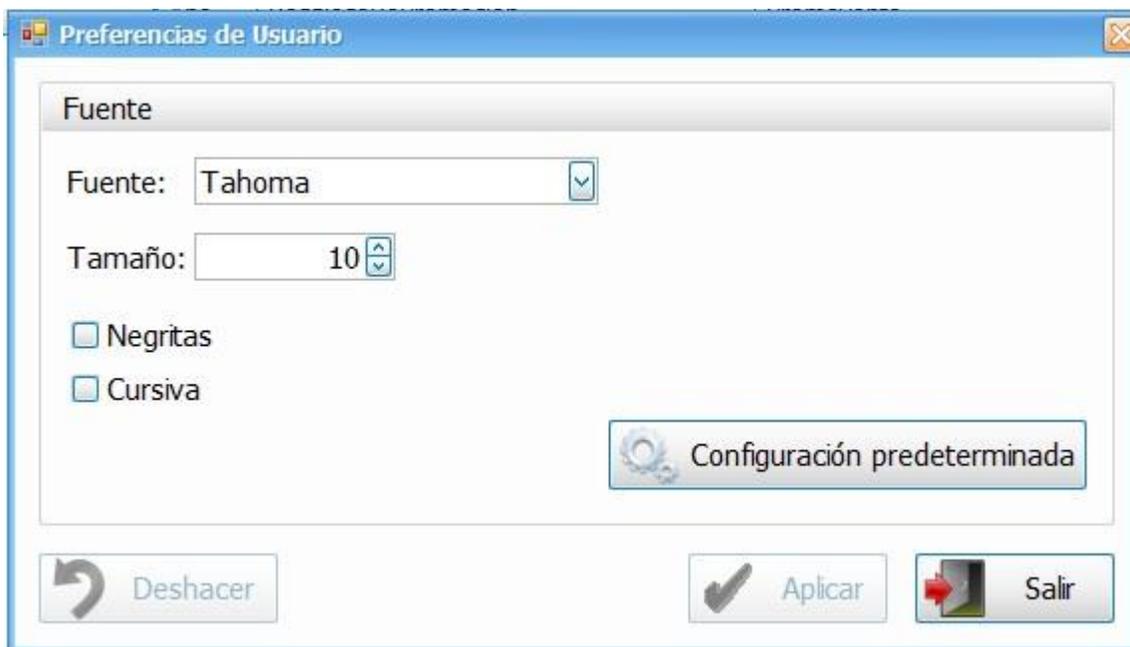
**Figura 4.3.8 Menú Edición con elementos desplegados**

Algunos elementos de éste menú, al elegirlos, desplegarán nuevas ventanas donde se seguirá el proceso para finalizar la acción que se necesite. Por ejemplo en la opción de abrir archivo, se abrirá una nueva ventana donde podremos elegir el archivo que queremos abrir, así como la posibilidad de modificarlo o de eliminarlo como se muestra en la figura 4.3.9.



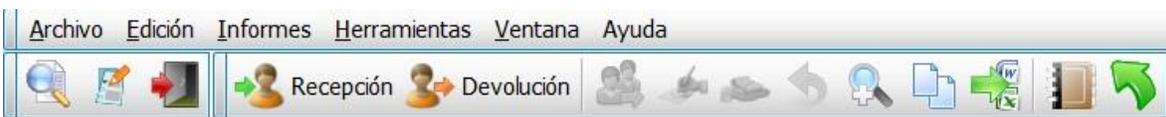
**Figura 4.3.9 Sub-ventana para la opción de abrir archivo.**

Otro ejemplo de esto es la sub-ventana de Preferencias en la pestaña de Herramientas del menú, donde se pueden elegir tipos de letra y tamaños para el área de trabajo. La figura 4.3.10 muestra ésta sub-ventana.



**Figura 4.3.10 Sub-ventana Preferencias.**

Debajo de este primer menú, se encuentra un menú gráfico. Este menú gráfico contiene algunos los elementos más utilizados en el menú principal y están representados por íconos para su fácil detección. El menú gráfico se creó para darle mayor dinamismo a la aplicación ya que con un simple clic se puede realizar las acciones más utilizadas. La figura 4.3.11 muestra el menú grafico, las opciones que ahí se encuentran son: búsqueda, seguimiento, cerrar aplicación, recepción de asuntos, devolución de asuntos, zoom, copiar, exportar, ficha del expediente y cerrar la ventana.



**Figura 4.3.11 Menú gráfico.**

En una tercera fila, existen los botones para elegir los parámetros de fecha, actualización de datos y un filtro donde podemos elegir el tipo de asuntos que queremos ver en nuestra área de trabajo. Además de que en esa misma fila, al momento de elegir un nuevo asunto podemos elegir también el secretario que se le debe de asignar a cada expediente. En el área de trabajo aparecerán los asuntos que están pendientes por turnar al siguiente proceso que es el de Acuerdo. En la figura 4.3.12 se muestran estos dos elementos.

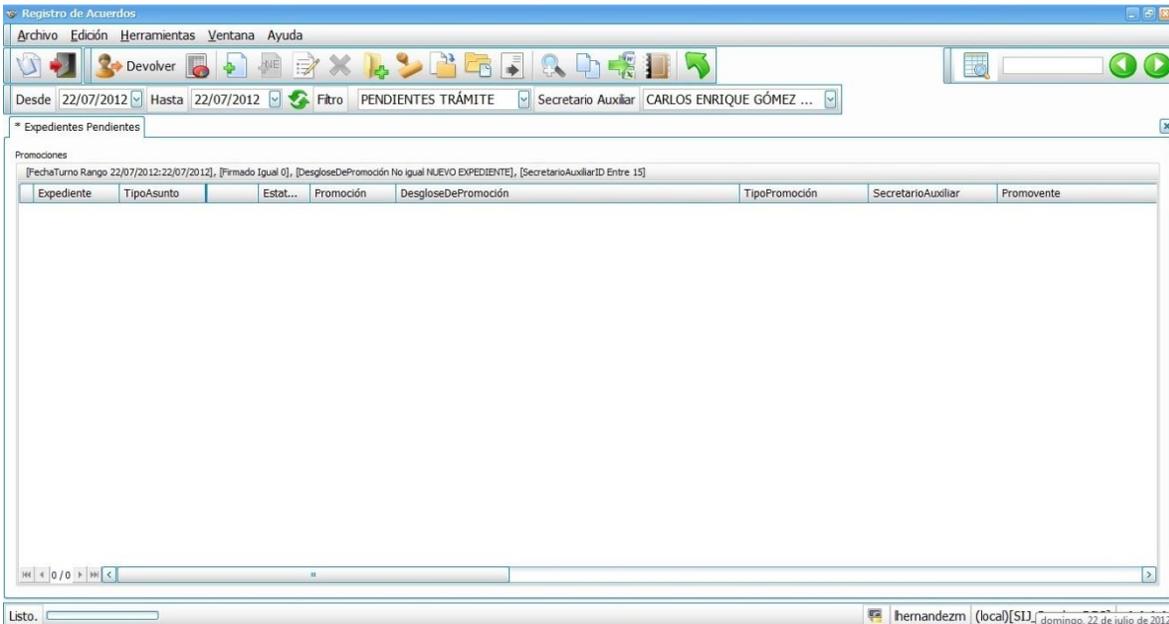


**Figura 4.3.12 Área de trabajo y menú de filtro de búsqueda.**

## Módulo de Acuerdos

Nuevamente, para este módulo, la primer pantalla que aparecerá al usuario será la de Login. Una vez verificado el usuario entrará a la pantalla principal del módulo.

Como en los módulos anteriores y para seguir un patrón en el diseño de la interfaz, la pantalla principal contendrá un menú principal con varias pestañas, enseguida, un menú gráfico con las principales acciones que se utilizan en el módulo, una barra para filtrar los documentos y por último el área de trabajo. En la figura 4.3.13 se muestra el diseño de esta ventana principal.



**Figura 4.3.13 Ventana principal del módulo Acuerdos.**

El menú principal cuenta con las siguientes pestañas:

- **Archivo:** Aquí se encuentran las opciones registro de acuerdos a promociones, exportar, actualizar, copiar, zoom, ficha del expediente, cerrar ventana y salir del programa.
- **Edición:** Las opciones aquí son las de devolver un expediente, dar de alta, modificar o eliminar un acuerdo, agregar una acuerdo a un expediente,

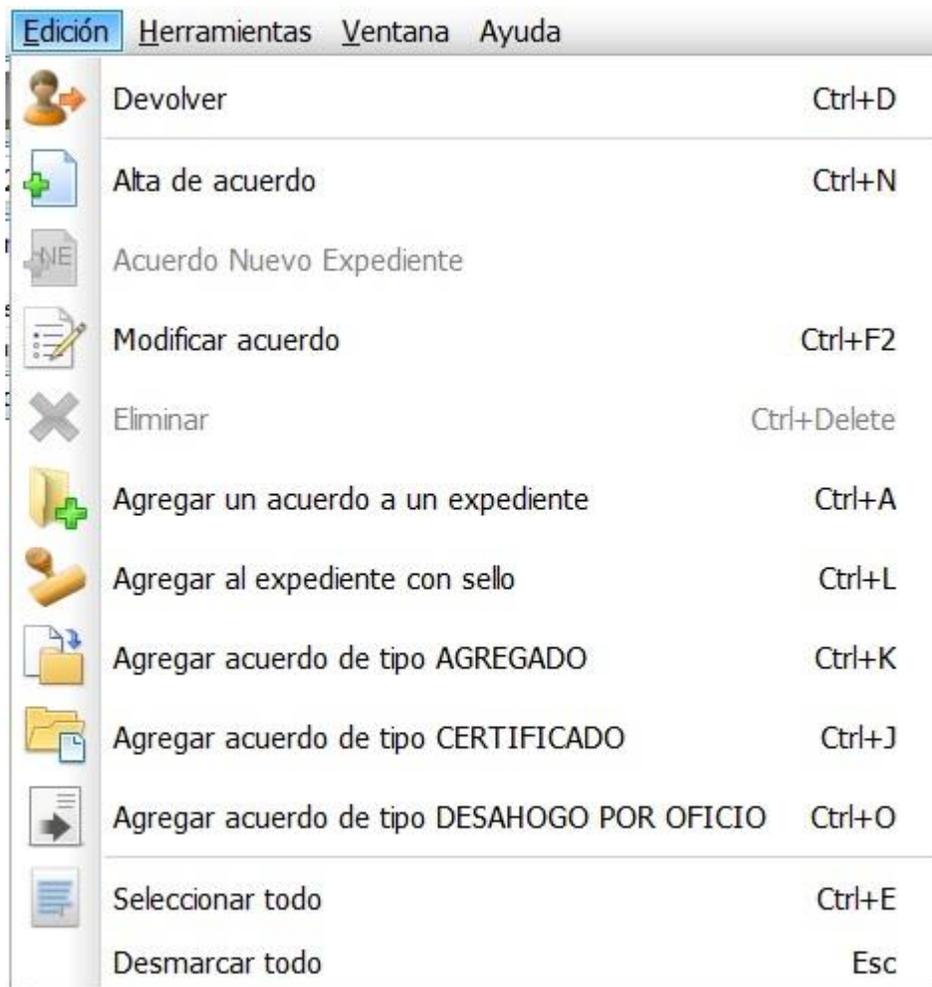
agregar al expediente con sello, 3 diferentes tipos de agregar acuerdos (agregado, certificado y desahogo por oficio), marcar y desmarcar texto.

- **Herramientas:** Aquí encontramos las opciones de configuración del programa, búsquedas de archivos o expedientes y organización de los registros en el área de trabajo.
- **Ventana:** Debido a que la aplicación puede tener varias instancias corriendo simultáneamente en la misma máquina, aquí se da la opción para organizar las instancias de una forma que sea útil al usuario.
- **Ayuda:** Como en todas las pantallas principales de nuestro sistema, existe éste botón para brindarle ayuda o una explicación de los componentes del sistema al usuario.

En la figuras 4.3.14 se muestra la pestaña del menú archivo desplegada, en la figura 4.3.15 se muestra la pestaña del menú edición desplegada.



**Figura 4.3.14 Pestaña Archivo desplegada.**



**Figura 4.3.15 Pestaña Edición desplegada.**

Algunos elementos de éste menú, al elegirlos, desplegarán nuevas ventanas donde se seguirá el proceso para finalizar la acción que se necesite. Por ejemplo, en la opción registro de acuerdo a promociones en la pestaña archivo del menú principal ó en la pestaña de edición en la opción alta acuerdo. En las figuras 4.3.16 y 4.3.17 se puede apreciar estas sub-ventanas respectivamente.

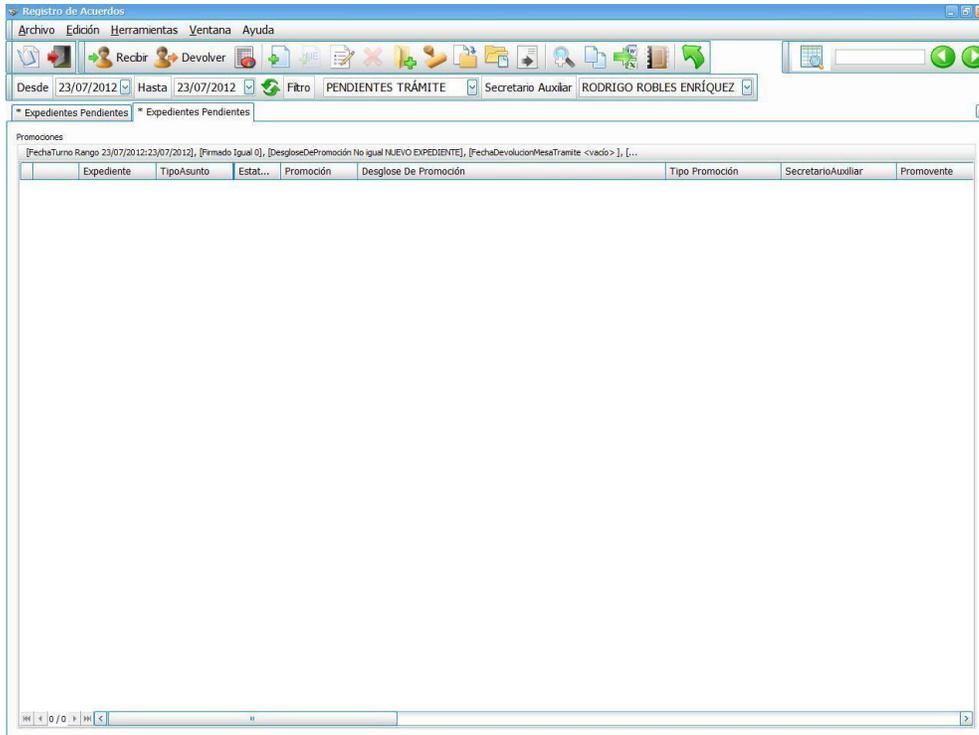


Figura 4.3.16 Sub-ventana “Registro de acuerdo”

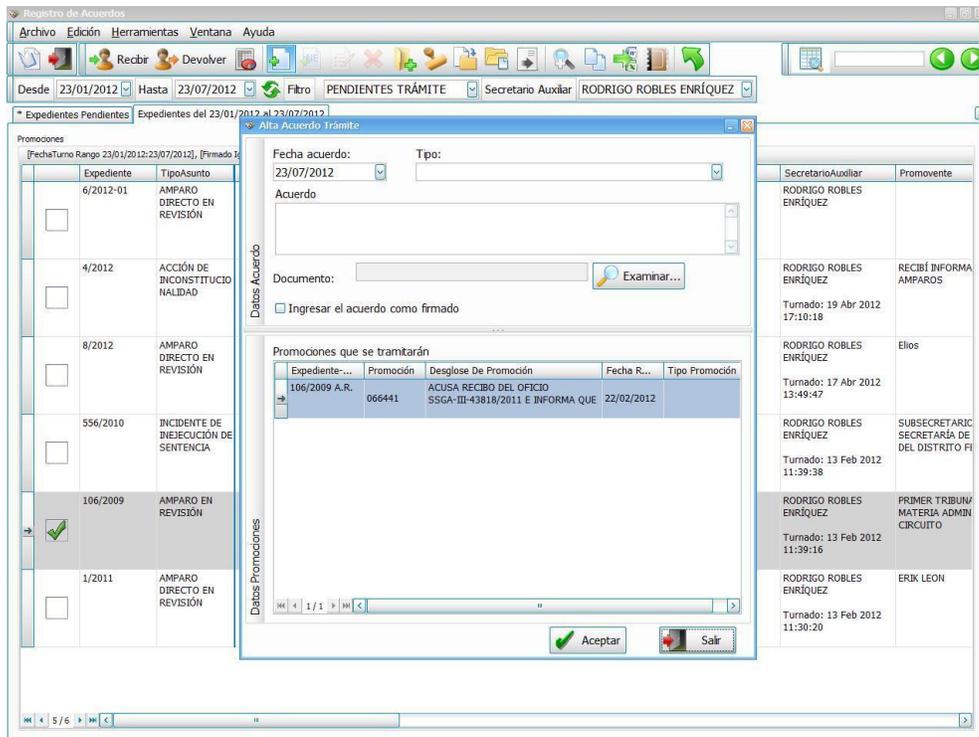


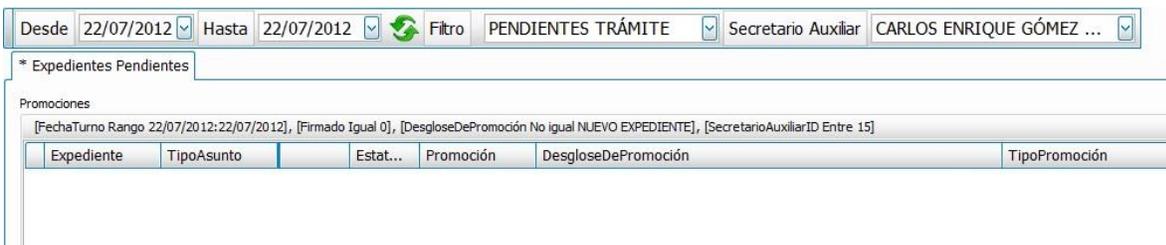
Figura 4.3.17 Sub-ventana “Alta acuerdo”

Debajo de este primer menú, se encuentra un menú gráfico. Este menú gráfico contiene algunos los elementos más utilizados en el menú principal y están representados por íconos para su fácil detección. El menú gráfico se creó para darle mayor dinamismo a la aplicación ya que con un simple clic se puede realizar las acciones más utilizadas. La figura 4.3.18 muestra el menú gráfico, las opciones que ahí se encuentran son: registro de acuerdo, salir de la aplicación, devolver, reporte estadístico, alta de acuerdo, acuerdo presidencial, modificar acuerdo, eliminar, agregar acuerdo (en las variantes antes mencionadas), aoom, copiar, exportar, ficha de expediente y cerrar ventana.



**Figura 4.3.18 Menú gráfico**

En una tercera fila, existen los botones para elegir los parámetros de fecha, actualización de datos y un filtro donde podemos elegir el tipo de asuntos que queremos ver en nuestra área de trabajo. Existe una menú desplegable que nos permite visualizar los expedientes mediante un filtro donde se agrupan por el proceso en el que se esté en ese momento. En el área de trabajo aparecerán los asuntos que están pendientes por turnar al siguiente proceso. En la figura 4.3.19 se muestran estos dos elementos.



**Figura 4.3.19 Área de trabajo y menú de filtro de búsqueda.**

## **4.4 Generación de pruebas y mantenimiento**

### **Integración**

La integración es una técnica sistemática para construir la estructura del programa o software, es decir, consiste en poner juntos a los módulos para detectar errores asociados con la interacción.

El sistema se compone de subsistemas formados por módulos que, a su vez, pueden componerse de procedimientos. Si se intenta probar el sistema como una sola entidad, es posible que no se identifique más de un pequeño porcentaje de errores. El proceso de pruebas debe avanzar por etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

### **Pruebas de software**

Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de computadora.

Básicamente es una fase en el desarrollo del software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases o etapas del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Para determinar el nivel de calidad, se deben efectuar medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

## **Tipos de pruebas**

- Pruebas unitarias
- Pruebas de Integración
- Pruebas de validación
- Pruebas de Caja Negra (Funcional).
- Pruebas de Caja Blanca (Estructural).
- Pruebas de regresión
- Pruebas de Estrés

### **Prueba unitaria**

En programación, una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto.

### **Pruebas de integración**

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

Las pruebas de integración es la fase de la prueba de software en la cual módulos individuales de software son combinados y probados como un grupo.

Son las pruebas posteriores a las pruebas unitarias y preceden a las pruebas del sistema.

### **Pruebas de validación**

Las pruebas de validación en la ingeniería de software son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido. La validación es el proceso de comprobar lo que se ha especificado, es lo que el usuario realmente quería.

Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.

### **Pruebas de Caja Negra (Funcional)**

Esta prueba se realiza con base a los requerimientos sin conocimiento sobre cómo fue construido el sistema y usualmente dirigidas a los datos, es decir se enfocan directamente en el exterior del módulo o componente, sin importar el código, son pruebas funcionales en las que se trata de encontrar fallas en las que no se atiene a su especificación, ya que éste es una caja negra y su comportamiento sólo puede ser determinado al estudiar sus entradas y las salidas relacionadas a éstas. Las pruebas son un enfoque complementario a las pruebas de caja blanca que intentan reconocer errores con las siguientes características:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores de estructura de datos o en accesos a bases de datos externas
- Errores de rendimiento

- Errores de inicio y terminación

### **Pruebas de Caja Blanca (Estructural)**

Este tipo de pruebas son el contraste al de la caja negra, debido a que quien realiza la prueba puede analizar el código, estas pruebas son mucho más amplias, normalmente se denominan pruebas de cobertura o pruebas de caja transparente, al total de pruebas de caja blanca se le llama cobertura, la cobertura es un número porcentual que indica cuánto código del programa se ha probado. Este tipo de prueba se realiza cuando se desarrolla un sistema, debido a que el programador conoce la lógica del sistema y la estructura de un componente para poder obtener los datos de prueba, conforme va realizando cualquier adecuación o desarrollo se corren pruebas para verificar la información, en muchas ocasiones, estas pruebas se realizan con ayuda de un usuario responsable del módulo del sistema en general, buscando:

- Garantizar que todas las trayectorias independientes dentro de un módulo hayan sido ejecutadas dentro de éste al menos una vez.
- Ejecutar todos los lados de las decisiones lógicas (verdaderas y falsas).
- Ejecutar todos los ciclos o bucles en sus límites operacionales.
- Ejecutar las estructuras de datos internas para asegurar su validez.

### **Pruebas de Regresión**

Cada vez que se añade un módulo nuevo como parte de las pruebas de integración, el software cambia: se establecen nuevos caminos en el flujo de datos, pueden existir nuevas I/O, y se invoca a una nueva lógica de control, lo cual puede ocasionar problemas con funciones que ya trabajaban correctamente. Las pruebas de regresión consisten en volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente, para asegurarse que los cambios no han ocasionado efectos colaterales indeseados.

Por otra parte todos los sistemas sufren una evolución a lo largo de la vida activa. En cada nueva versión se corrigen defectos o se añaden nuevas funciones, o ambas cosas. En cualquier caso, una nueva versión exige una nueva verificación.

Si éstas se han sistematizado en una fase anterior, ahora pueden volver a realizarse automáticamente, simplemente para comprobar que las modificaciones no provocaron errores donde antes no los había.

Por consiguiente, esta prueba detectará fallas en algún módulo debido a modificaciones realizadas con anterioridad.

### **Pruebas de Estrés**

Son utilizadas normalmente para someter a la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios muy superior al esperado.

### **Pruebas realizadas al sistema**

#### **Pruebas Unitarias**

#### **Búsqueda en el grid**

La funcionalidad de búsqueda de palabras en el grid es sensible a mayúsculas y minúsculas, así como acentos y demás signos de puntuación. Cuando encuentra una coincidencia pinta en color amarillo la celda donde se encontró la palabra. Ver figura 4.4.1.

FechaR...	FechaR...	SecretarioAuxiliar	ActoReclamado	Observacion
04/09/2012		CARLOS ENRIQUE GÓMEZ MARTÍNEZ Turnado: 04 Sep 2012 11:54:24	pruebas	
19/07/2012		FRANCISCO JAVIER QUIÑONES PEÑA Turnado: 19 Jul 2012 20:17:51		

**Figura 4.4.1 Búsqueda sobre el grid**

### Búsqueda rápida

La búsqueda rápida es una funcionalidad para encontrar promociones, en base al número de expediente, tipo de asunto y opcionalmente el número de registro. Ver figura 4.4.2.

Búsqueda Rápida de Expediente

Expediente [Consecutivo / Año - Extensión]  
2 / 2013 -

Tipo de Asunto  
AMPARO EN REVISIÓN

Número de Registro  
000011

Aceptar Salir

**Figura 4.4.2 Ventana de búsqueda rápida**

La búsqueda se hace directamente a la base de datos, abriendo una nueva pestaña con los resultados. Ver figura 4.4.3.

Registro de Acuerdos

Archivo Edición Herramientas Ventana Ayuda

Desde [Vacio] Hasta [Vacio] Filtro [Vacio] Secretario Auxiliar [Vacio]

Expedientes del 28/04/2012 al 28/04/2013 \* Búsqueda rápida de expediente

Promociones

[TipoAsuntoID Igual AMPARO EN REVISIÓN], [Promoción Entre 000011], [Consecutivo Igual 2], [Año Igual 2013], [AsuntoPertenenciaID Igual 3]

Expediente	Tipo Asunto	Detalles importantes	Promoción	Desglose De Promoción
2/2013	AMPARO EN REVISIÓN	Promoción turnada el 23 Ene 2013 17:14:09 Promoción sin acuerdo Área: SUBSECRETARÍA GENERAL DE ACUERDOS Filtro - PENDIENTES TRÁMITE	000011	segunda promoción

**Figura 4.4.3 Resultados de la búsqueda rápida**

### Búsqueda mediante filtro avanzado

Este tipo de búsqueda es para cuando se requieren un mayor número de criterios, haciendo que los resultados sean más específicos. Ver figura 4.4.4.

Filtros

Filtro avanzado

**IMPORTANTE: Mientras más específicos sean los criterios el tiempo de respuesta es menor**

Ingrese los criterios de búsqueda:

Campo	Operador	Criterio	...
Expediente	Igual	1/2012	✖
Fecha Turno	Rango	28/04/2010:28/04/2013	✖
Area	Entre	PLENO	✖
Desglose De Promoción	<no vacío>		✖
*			✖

Ingrese los criterios de ordenamiento:

Campo	Ordenamiento	...
Fecha Recepción	Ascendente	✖
*		✖

**Figura 4.4.4 Ventana para filtros avanzados**

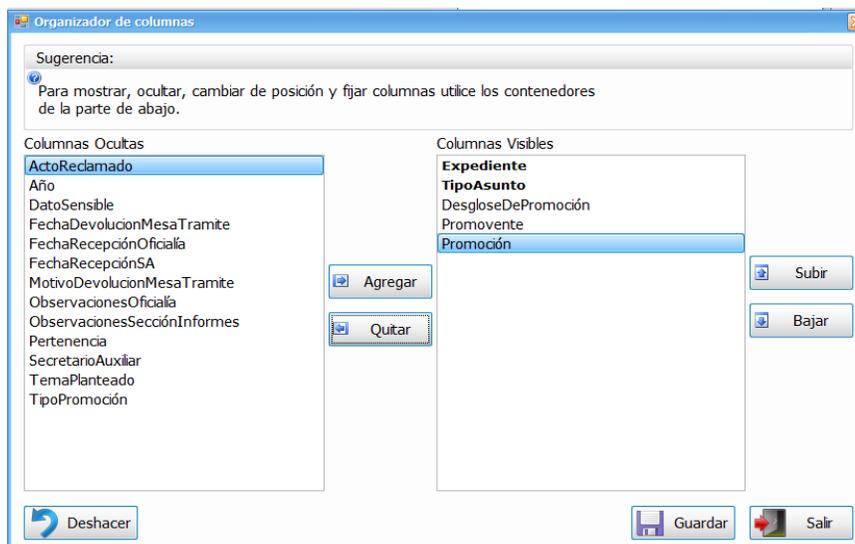
Los resultados se muestran en una nueva pestaña. Ver figura 4.4.5.

Promociones del 28/04/2012 al 28/04/2013		Nuevo Filtro				
	Expediente	TipoAsunto	Pertenencia	FechaR...	FechaR...	SecretarioAuxiliar
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	PLENO	19/07/2012		
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	PLENO	19/07/2012		FRANCISCO JAVIER QUIÑONES PEÑA Turnado: 19 Jul 2012 20:17:51
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	PLENO	19/07/2012		CARLOS ENRIQUE GÓMEZ MARTÍNEZ Turnado: 19 Jul 2012 17:53:03
<input type="checkbox"/>	1/2012	AMPARO DIRECTO EN REVISIÓN	PLENO	04/09/2012		CARLOS ENRIQUE GÓMEZ MARTÍNEZ Turnado: 04 Sep 2012 18:48:00

**Figura 4.4.5 Resultados del filtro avanzado**

## Organizar columnas

Con esta ventana se puede manipular el orden y las columnas que aparecerán en el grid, esto es útil ya que posteriormente se pueden crear reportes donde se muestren sólo los datos necesarios. Ver figuras 4.4.6 y 4.4.7.



**Figura 4.4.6 Ventana para organizar columnas**

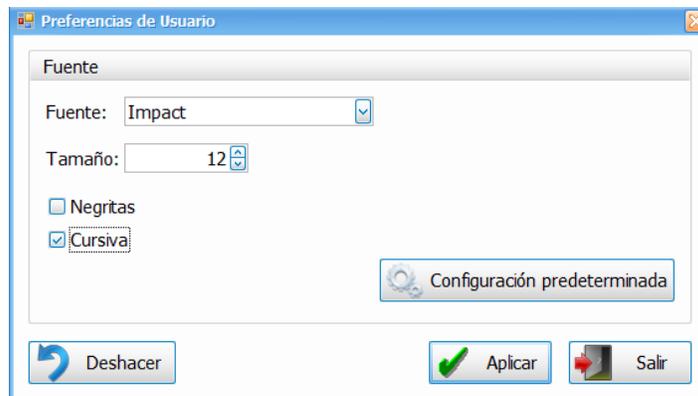
Promociones del 28/04/2012 al 28/04/2013

	Expediente	TipoAsunto	DesgloseDePromoción	Promoviente	Promoción
<input type="checkbox"/>	2/2012	AMPARO DIRECTO	NUEVO EXPEDIENTE		121212
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	Pruebas chelito		000002
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	Pruebas chelito		000003

**Figura 4.4.7 Resultado de organizar las columnas**

## Preferencias

Esta funcionalidad es únicamente de estilo, pudiendo modificar el tipo de letra, tamaño, remarcado y modo cursiva. Ver figuras 4.4.8 y 4.4.9.



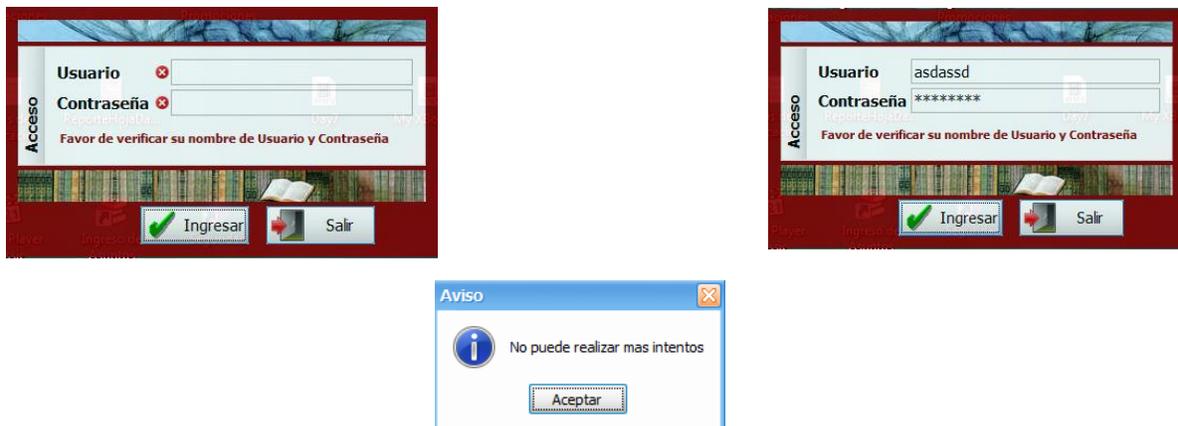
**Figura 4.4.8 Ventana para modificar las preferencias**

	<i>Expediente</i>	<i>TipoAsunto</i>	<i>DesgloseDePromoción</i>	<i>Promovente</i>	<i>Promoción</i>
<input type="checkbox"/>	2/2012	AMPARO DIRECTO	NUEVO EXPEDIENTE		121212
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	Pruebas chelito		000002
<input type="checkbox"/>	1/2012	AMPARO DIRECTO	Pruebas chelito		000003

**Figura 4.4.9 Resultado de modificar las preferencias**

### Pruebas de validación

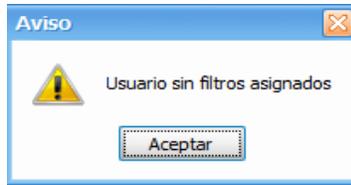
La primera validación es a nivel aplicación, marca con una x los campos que son requeridos. La segunda es a nivel de base de datos, autentica que los datos de usuario y contraseña sean los correctos. Sólo permite 3 intentos y después muestra la pantalla indicando que ya no puede realizar más. Ver figura 4.4.10.



**Figura 4.4.10 Pruebas ventana acceso**

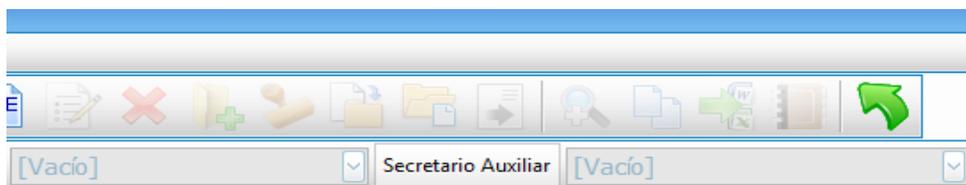
### Prueba de asignación de filtros

Después de validar el usuario y la contraseña se prosigue a validar que tenga asignados los filtros correspondientes a ese módulo. En caso de no ser así, muestra la siguiente pantalla, ver figura 4.4.11:



**Figura 4.4.11 Ventana usuario sin filtro**

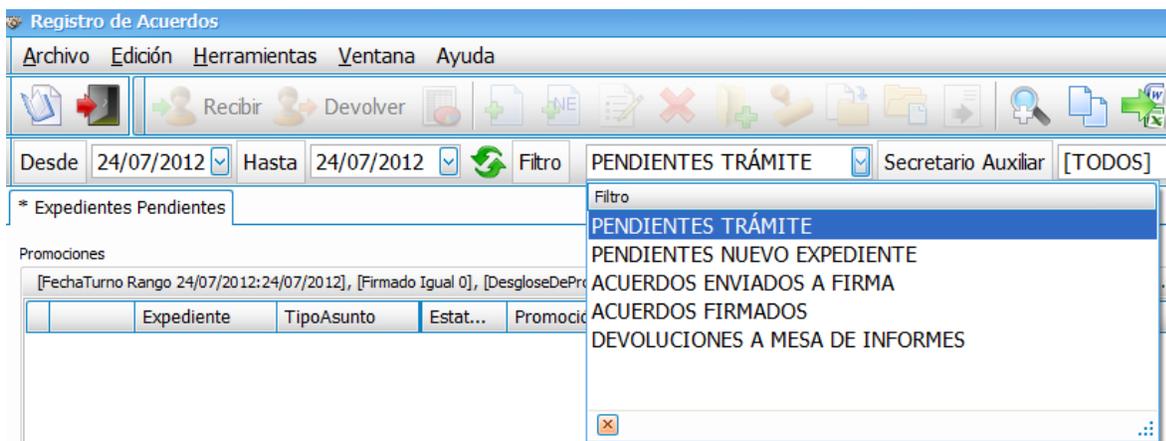
Para después deshabilitar los botones de la barra de herramientas, ver figura 4.4.12:



**Figura 4.4.12 Barra de herramientas deshabilitada**

### Pruebas de validación de perfil

Aunque el usuario tenga asignado los filtros correctos, si no tiene asignado ningún perfil no podrá realizar ninguna de las acciones principales de cada módulo. Usará el módulo sólo para consulta de información, ver figura 4.4.13:



**Figura 4.4.13 Filtros del sistema**

## Pruebas de integración

Para comprobar el correcto flujo y comunicación entre los distintos módulos se realizarán todos los pasos que debe seguir un expediente, empezando por darlo de alta en Ingreso Asuntos.

Son cuatro los campos requeridos para realizar esta operación: Tipo de Asunto, Expediente, Fecha de creación y número de registro. Algunos son de sólo lectura, que se cargan conforme al tipo de asunto que se elige y a los datos del usuario firmado en el equipo, ver figura 4.4.14:

Alta de expediente

\* Tipo de Asunto: ACLARACIÓN DE SENTENCIA

Subtipo de Asunto: <Elija una opción>

\* Expediente: 3 / 2012 - Ext. \* Fecha y hora del reloj fechador: 24/07/2012 13:56

Órgano de radicación: <Elija una opción>

Ubicación física (Destino): SUBSECRETARÍA GENERAL DE ACUERDOS

\* Número de registro: 121212

Número de Expediente Único Nacional: [Empty]

Número de Oficio o Escrito: [Empty]

Estado de la federación: <Elija una opción>

Municipio: [Vacío]

Materia y SubMateria:

- ADMINISTRATIVA
- AGRARIA
- FISCAL
- ELECTORAL
- TRATADOS INTERNACIONALES
- ASCASCAS
- CIVIL
- ELECTORAL
- FAMILIAR
- TRATADOS INTERNACIONALES
- COMÚN
- ELECTORAL
- TRATADOS INTERNACIONALES
- CONSTITUCIONAL
- ELECTORAL
- TRATADOS INTERNACIONALES
- DE TRABAJO
- ELECTORAL
- TRATADOS INTERNACIONALES
- PENAL
- ELECTORAL
- TRATADOS INTERNACIONALES

Cuadernos y Anexos: [Empty]

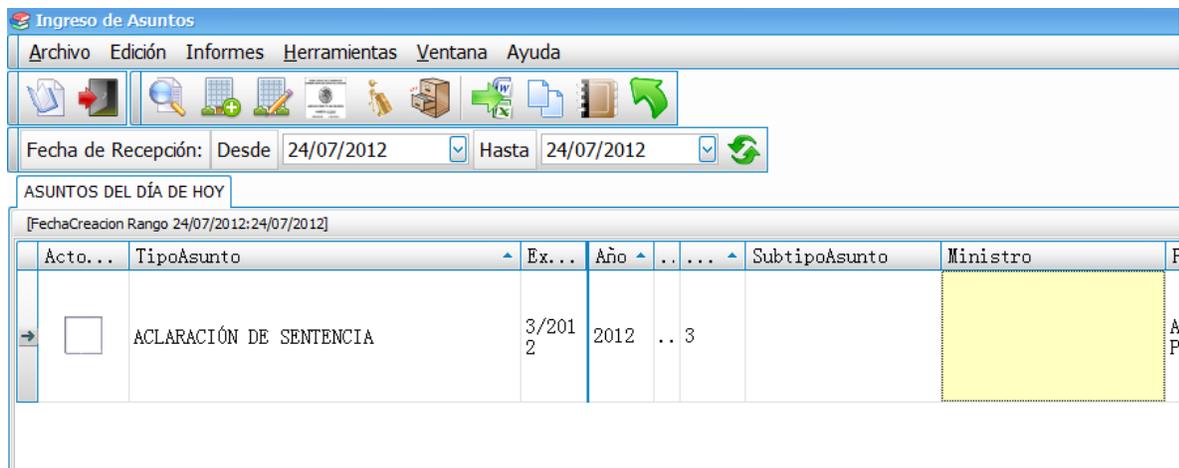
Acto Reclamado: [Empty]

Garantías violadas: [Empty]

Derechos humanos relacionados: [Empty]

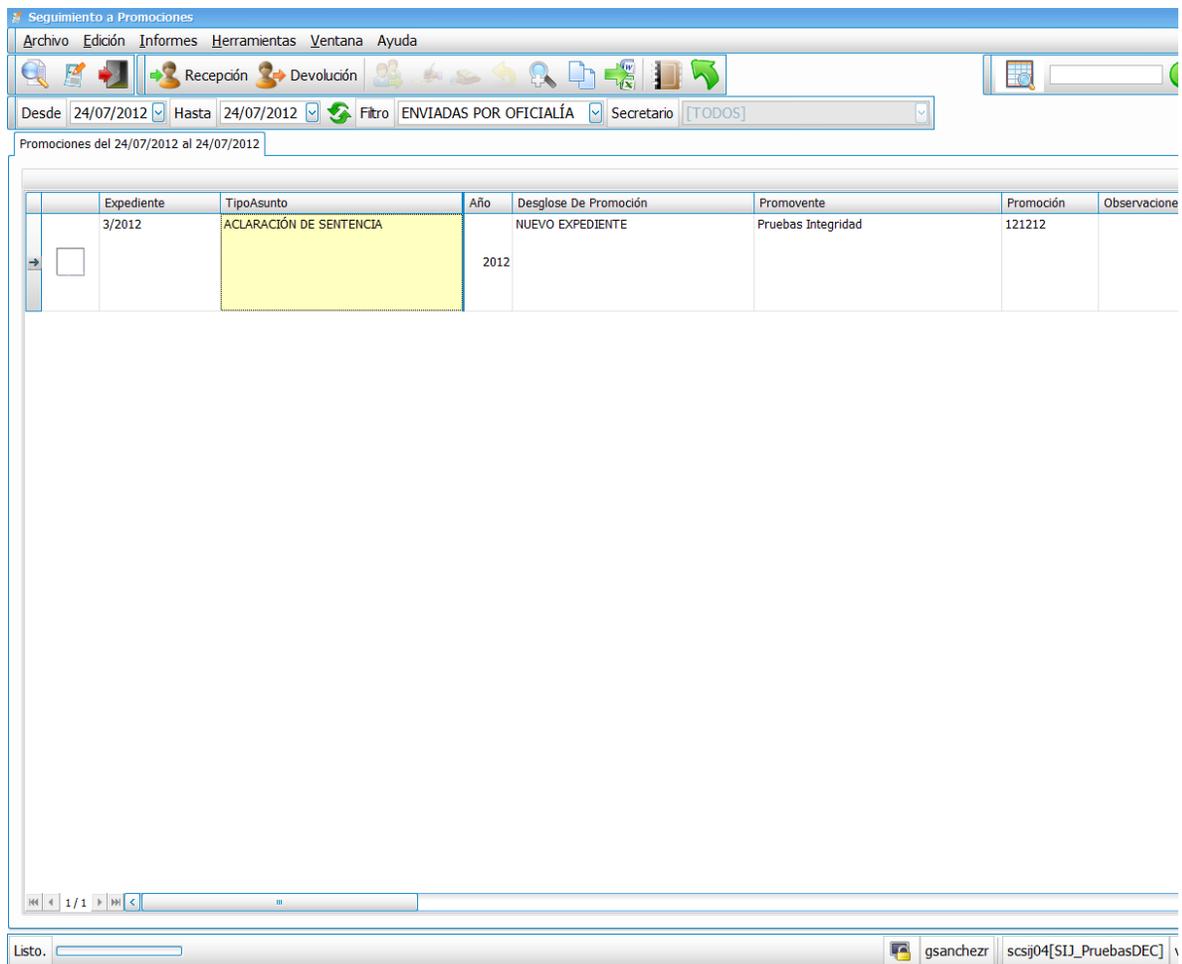
Figura 4.4.14 Ventana alta de expediente

Una vez dado de alta se cierra esa ventana automáticamente y el nuevo registro aparecerá en la ventana principal, ver figura 4.4.15:



**Figura 4.4.15 Ventana principal del módulo Ingreso de Asuntos**

Al dar de alta un nuevo expediente se genera automáticamente un registro en la tabla Promocion, el cual debe aparecer en el módulo Seguimiento a Promociones en el filtro Enviadas por Oficialía, ver figura 4.4.16:



**Figura 4.4.16 Ventana principal de Seguimiento a Promociones**

Cuando se dio de alta el expediente, en el campo Ubicación física (destino) se eligió Subsecretaría General de Acuerdos. En Seguimiento a Promociones para que un usuario pueda ver la promoción que se genera, debe pertenecer a dicha área.

La principal operación de este módulo es asignar las promociones a un secretario auxiliar para que elabore el acuerdo correspondiente. Con el botón Recepción se despliega la pantalla para realizar ésta acción, ver figura 4.4.17:

Recepción y turno de promociones

**Tipo Asunto**  
 ACLARACIÓN DE SENTENCIA

**Expediente**      **Órgano de radicación:**  
 3/2012      PLENO

Folio  
 121212

Promoción  
 NUEVO EXPEDIENTE

Observaciones de Oficialía de Partes

Secretario Auxiliar  
 [ASIGNE UN SECRETARIO AUXILIAR]

Fecha Recepción      Fecha de Turno a Secretario Auxiliar  
 24/07/2012 02:29 p.m.      24/07/2012 02:29 p.m.

Descripción

Observaciones

Aceptar      Salir

**Figura 4.4.17 Ventana de Recepción de promoción**

Se muestran algunos datos importantes del expediente y se deberá elegir el secretario auxiliar y las fechas de recepción y turno de la promoción, opcionalmente se podrán introducir algunas observaciones de dicha acción.

Si la operación se realiza con éxito, la promoción pasará al filtro Promociones Turnadas, ver figura 4.4.18.

Expediente	TipoAsunto	Año	Desglose De Promoción	Promoviente	Promoción	Observaciones
3/2012	ACLARACIÓN DE SENTENCIA	2012	NUEVO EXPEDIENTE	Pruebas Integridad	121212	Pruebas integ
31/2012	AMPARO DIRECTO EN REVISIÓN	2012	NUEVO EXPEDIENTE		343535	

**Figura 4.4.18 Filtro de Promociones Turnadas**

En el módulo Registro de Acuerdos también aparecerá dicha promoción, en el filtro Pendientes Nuevo Expediente, ver figura 4.4.19.

Expediente	TipoAsunto	Estat...	Promoción	Desglose De Promoción	Tipo Promoción	SecretarioAuxiliar	Promove
3/2012	ACLARACIÓN DE SENTENCIA	<input checked="" type="checkbox"/>	121212	NUEVO EXPEDIENTE	Nuevo Expediente	RODRIGO ROBLES ENRÍQUEZ Turnado: 24 Jul 2012 14:29:01	Pruebas
31/2012	AMPARO DIRECTO EN REVISIÓN	<input type="checkbox"/>	343535	NUEVO EXPEDIENTE	Nuevo Expediente	RODRIGO ROBLES ENRÍQUEZ Turnado: 24 Jul 2012 13:27:50	

**Figura 4.4.19 Ventana principal de Registro de Acuerdos**

La función principal de este módulo es realizar los acuerdos de las promociones, para ello existen varias formas de hacerlo. La forma más general será con el botón Alta de Acuerdo, el cual mostrará la siguiente ventana, ver figura 4.4.20:

Alta Acuerdo Nuevo Expediente

Fecha Acuerdo: 24/07/2012 Tipo:

Acuerdo

Documento:  Examinar... 

Ingresar el acuerdo como firmado

Datos Promociones

Promociones que se tramitarán

Expediente-...	Promoción	Desglose De Promoción	Fecha R...	Tipo Promoción
3/2012 → ACLARA.SENT.	121212	NUEVO EXPEDIENTE	24/07/2012	Nuevo Expedient

1 / 1

Aceptar Salir

**Figura 4.4.20 Ventana de alta de acuerdo**

En la parte inferior se muestra la promoción a la cual se le realizará el acuerdo, con algunos datos importantes. Los campos Tipo y Acuerdo son requeridos y con el botón Examinar se podrá seleccionar el documento que se adjuntará al acuerdo.

Si la operación se realiza con éxito, el acuerdo se podrá visualizar en el filtro Acuerdos Enviados a Firma, ver figura 4.4.21.

Registro de Acuerdos

Archivo Edición Herramientas Ventana Ayuda

Recibir Devolver

Desde 24/07/2012 Hasta 24/07/2012 Filtro ACUERDOS ENVIADOS A... Secretario Auxiliar RODRIGO ROBLES ENRÍQUEZ

Acuerdos del 24/07/2012 al 24/07/2012

Acuerdos

[FechaAcuerdo <no vacío> ], [Firmado Igual 0], [Devuelto Igual 0], [SecretarioAuxiliarID Entre 97], [FechaAcuerdo Rango 24/07/2012:24/07/2012], [Secre...

	Expediente	TipoAsunto	Acuerdo	Tipo Acuerdo	Fecha Acuerdo	SecretarioAuxiliar	Documento	Folio
<input type="checkbox"/>	3/2012	ACLARACIÓN DE SENTENCIA	Pruebas integración	DEVOLUCIÓN AL TRIBUNAL COLEGIADO POR INCOMPETENCIA	24/07/2012	RODRIGO ROBLES ENRÍQUEZ		121212
<input type="checkbox"/>	26/2012	AMPARO DIRECTO	capacitación	DESECHAMIENTO	24/07/2012	RODRIGO ROBLES ENRÍQUEZ		038371
<input type="checkbox"/>	33/2012	AMPARO DIRECTO EN REVISIÓN	test	ADMISIÓN Y TURNO	24/07/2012	RODRIGO ROBLES ENRÍQUEZ		123

Promociones

Expediente	TipoAsunto	Estat...	Promoción	Desglose De Promoción	Tipo Promoción	SecretarioAuxiliar	Observaciones Sec
26/2012	AMPARO DIRECTO		038371	NUEVO EXPEDIENTE	Nuevo Expediente	RODRIGO ROBLES ENRÍQUEZ	

**Figura 4.4.21 Filtro de Acuerdos Enviados a Firma**

En este filtro la ventana se encuentra dividida en dos, en la parte de arriba se muestran los acuerdos y en la parte de abajo las promociones relacionadas al acuerdo seleccionado. Existe una columna con el nombre Documento la cual en realidad es un botón, si al acuerdo se le adjunto un archivo este botón se muestra habilitado y al presionarlo abre en primer plano dicho archivo.

En el módulo de Seguimiento a Promociones existe el mismo filtro Acuerdos Enviados a Firma. Desde este módulo se realiza la última operación en el ciclo de vida del expediente, la cual es firmar el acuerdo, ver figura 4.4.22.

Seguimiento a Promociones

Archivo Edición Informes Herramientas Ventana Ayuda

Recepción Devolución

Desde 24/07/2012 Hasta 24/07/2012 Filtro ACUERDOS ENVIADOS A ... Secretario [TODOS]

Acuerdos del 24/07/2012 al 24/07/2012

Expediente	TipoAsunto	ActoReclamado	ActoReclamado	ActoReclamado	SecretarioAuxiliar	ActoReclam...	Folio
3/2012	ACLARACIÓN DE SENTENCIA	Pruebas integración	DEVOLUCIÓN AL TRIBUNAL COLEGIADO POR INCOMPETENCIA	24/07/2012	RODRIGO ROBLES ENRÍQUEZ		121212
26/2012	AMPARO DIRECTO	capacitación	DESSEGUIMIENTO	2012	RODRIGO ROBLES ENRÍQUEZ		038371
33/2012	AMPARO DIRECTO EN REVISIÓN	test		24/07/2012	RODRIGO ROBLES ENRÍQUEZ		123

Firmar acuerdo

¿Desea calificar el acuerdo como FIRMADO?

Sí No

Expediente	TipoAsunto	Año	Desglose De Promoción	Promovente	Promoción	Observaciones Sección
3/2012	ACLARACIÓN DE SENTENCIA	2012	NUEVO EXPEDIENTE	Pruebas Integridad	121212	Pruebas integración

**Figura 4.4.22 Ventana para firmar acuerdo**

Al confirmar esta operación, el acuerdo pasará al filtro final de todo este ciclo, Acuerdos Firmados, ver figura 4.4.23:

Seguimiento a Promociones

Archivo Edición Informes Herramientas Ventana Ayuda

Recepción Devolución

Desde 24/01/2012 Hasta 24/07/2012 Filtro ACUERDOS FIRMADOS Secretario [TODOS]

Acuerdos del 24/01/2012 al 24/07/2012

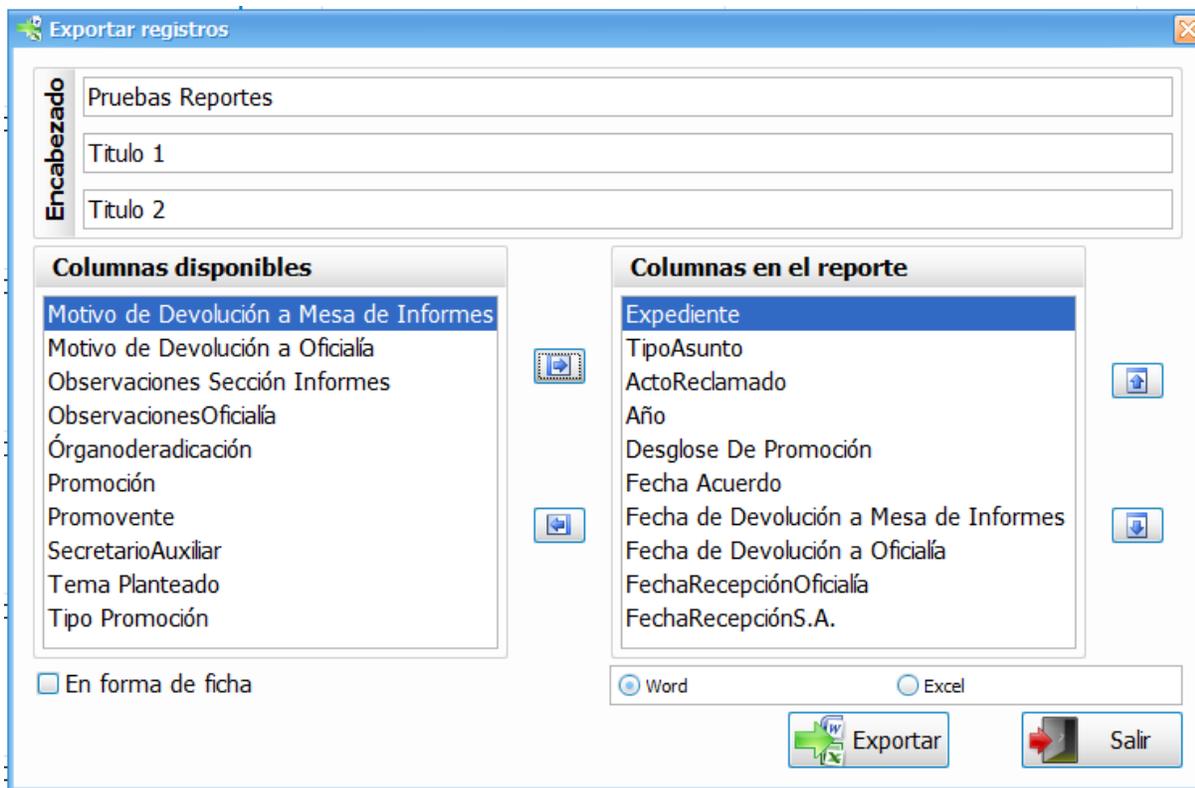
	Expediente	TipoAsunto	ActoReclamado	ActoReclamado	ActoReclamado	SecretarioAuxiliar	ActoReclam...	Folio
<input type="checkbox"/>	24/2012	AMPARO DIRECTO EN REVISIÓN	Prueba de Devolución OBEM	TRÁMITE (SE ORDENÓ FORMAR RECURSO DE APELACIÓN)	17/07/2012	ARTURO HERNÁNDEZ GILES		526351
<input type="checkbox"/>	23/2012	AMPARO DIRECTO	síntesis del AUTO INICIAL en subse	ADMISIÓN Y TURNO	09/07/2012	RODRIGO ROBLES ENRÍQUEZ		928367
<input type="checkbox"/>	1/2008	AMPARO DIRECTO	Promoción agregada al expediente solo con sello	CON SELLO	02/07/2012	RODRIGO ROBLES ENRÍQUEZ		749322

1 / 37

**Figura 4.4.24 Filtro acuerdos firmados**

## 4.5 Generación de Reportes

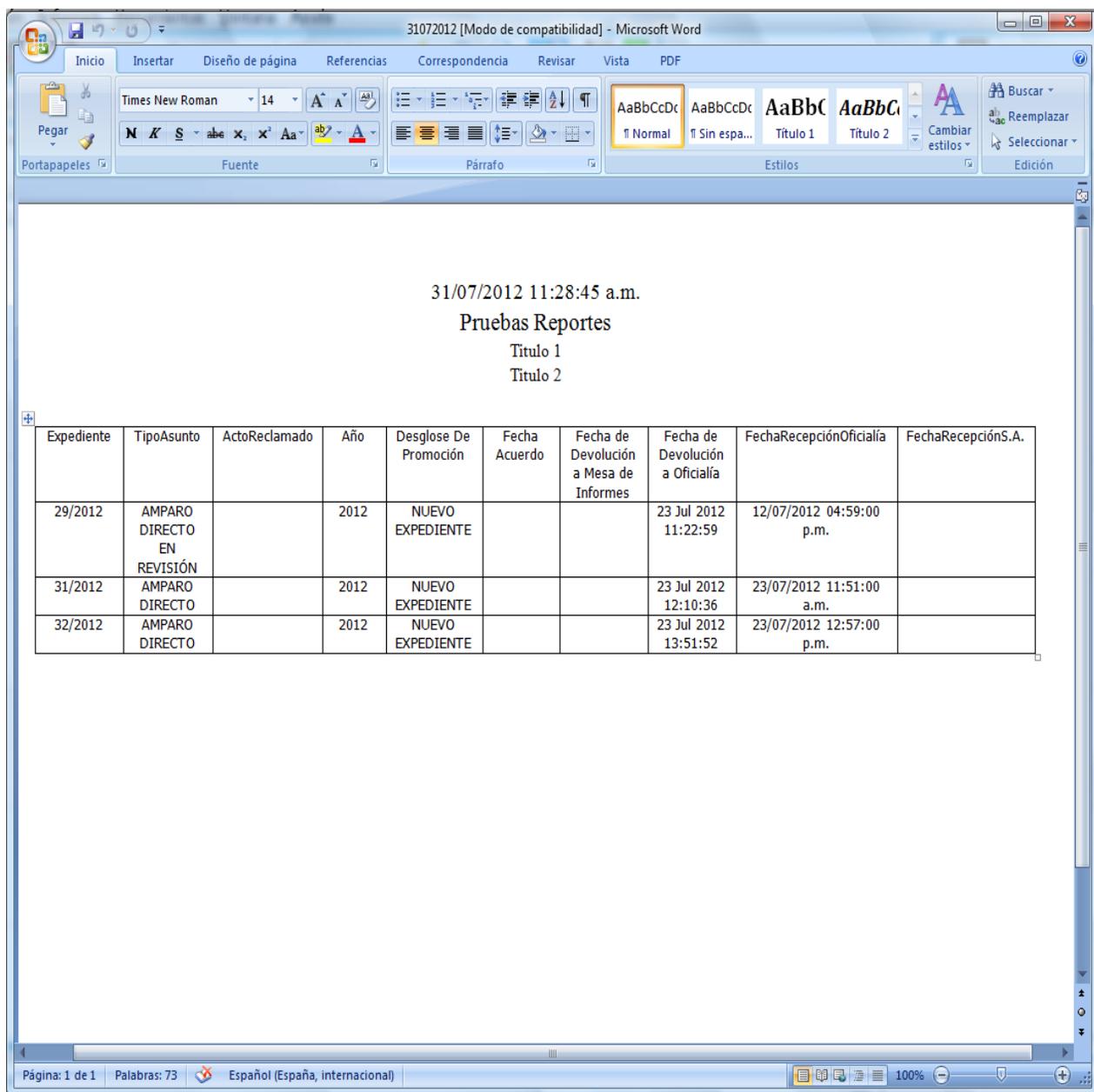
Uno de los aspectos que resultaban más tardados para los usuarios antes del sistema era la generación de reportes, capturar los expedientes en documentos electrónicos y organizarlos según su tipo y fecha. Es por ello que se creó la funcionalidad de exportar los registros a un documento tipo Word o Excel, según se requiera. Ver figura 4.5.1



**Figura 4.5.1 Ventana para exportar registros**

En la figura 4.5.1 se muestra la pantalla que sirve para elegir en primera instancia los encabezados que llevará el documento, después las columnas que queremos que se muestren en dicho documento teniendo la opción de organizarlas por el orden que se quiera, y al final se elige si será en formato Word o Excel.

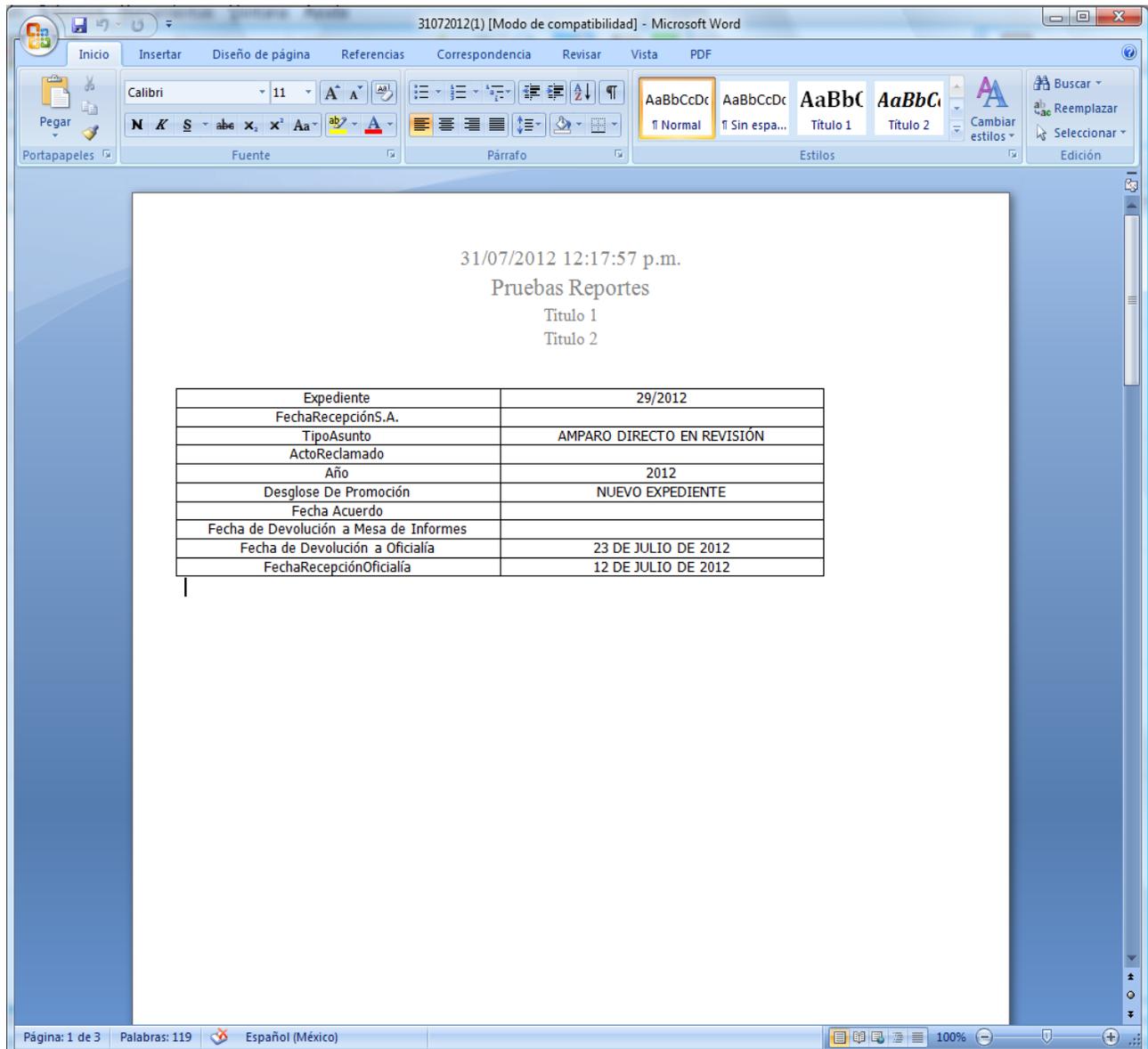
En la figura 4.5.2 se puede ver el documento que genera:



**Figura 4.5.2 Documento con los registros exportados**

Hasta arriba se muestra la fecha y hora de generación del documento, seguido por los encabezados que introducimos y después los registros con las columnas que elegimos.

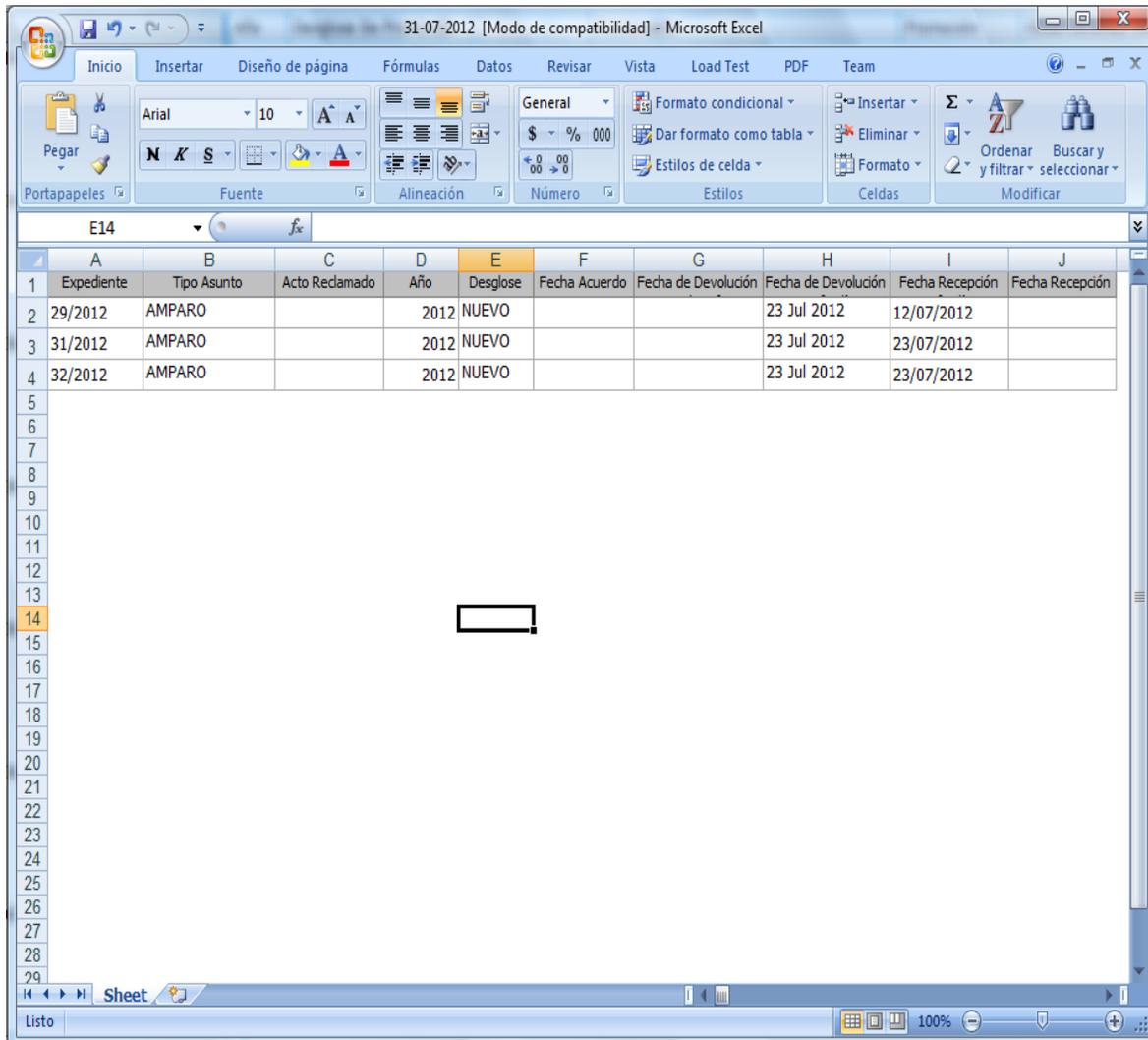
Existe otra forma de generar el documento en Word, llamada forma de ficha. Se selecciona el control tipo checkbox que lleva la etiqueta “En forma de ficha” y generará el siguiente reporte. Ver figura 4.5.3



**Figura 4.5.3 Registros exportados en forma de ficha**

La parte superior es igual al formato anterior, lo único que cambia es la forma que muestra los registros, con el nombre de las columnas en forma de lista y a la derecha el valor que tienen, cada registro en una página distinta.

En Excel sólo tiene un único formato, el cual es como se muestra en la figura 4.5.4:



	A	B	C	D	E	F	G	H	I	J
1	Expediente	Tipo Asunto	Acto Reclamado	Año	Desglose	Fecha Acuerdo	Fecha de Devolución	Fecha de Devolución	Fecha Recepción	Fecha Recepción
2	29/2012	AMPARO		2012	NUEVO			23 Jul 2012	12/07/2012	
3	31/2012	AMPARO		2012	NUEVO			23 Jul 2012	23/07/2012	
4	32/2012	AMPARO		2012	NUEVO			23 Jul 2012	23/07/2012	
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										

**Figura 4.5.4 Registros exportados a documento de tipo Excel**

Como se puede ver en la figura 4.5.4 las columnas en muestran con un fondo gris en el orden en que se seleccionaron, y el valor de los registros por debajo de éstas.

Para el módulo de Registro de Acuerdos se tiene un botón extra con la funcionalidad de Reporte Estadístico. Muestra el total de promociones y acuerdos relacionados al secretario auxiliar firmado en el sistema, organizados por tipos y tomando en cuenta el rango de fechas seleccionadas.

La figura 4.5.5 muestra la ventana de dicho reporte estadístico:

Descripción	Fecha Desde	Fecha Hasta
RODRIGO ROBLES ENRÍQUEZ	01/01/2012	31/07/2012
Promociones de trámite turnadas	41	
Promociones de trámite pendientes de acuerdo	5	
Promociones de nuevo expediente turnadas	66	
Promociones de nuevo expediente pendientes de auto de presidencia	35	
Promociones devueltas	31	
Total de acuerdos que desahogan promociones	59	
AVOCAMIENTO	2	
TRAMITE	1	
FALLO	1	
TURNO A PONENTE	1	
AUTO INICIAL	1	
COMPETENCIA DELEGADA	1	
RADICACIÓN	1	
CON SELLO	5	
AGREGADO	2	
PREVENCIÓN	1	
ACLARACIÓN DE DEMANDA	5	
ADMISIÓN DE LA DEMANDA	2	
DESECHAMIENTO DE DEMANDA	1	
CONTESTACIÓN DE DEMANDA	1	
ADMISIÓN DE LA AMPLIACIÓN DE DEMANDA	3	
PEDIMENTO PGR	1	
OPINIÓN DE PGR	1	
CONCEDE SUSPENSIÓN	1	
NIEGA SUSPENSIÓN	1	
DESAHOGA VISTA	1	
CUMPLE REQUERIMIENTO	1	
CONTESTA AMPLIACIÓN DE DEMANDA	1	
FORMULA ALEGATOS	2	
CONTESTA DEMANDA Y RECONVIENE AL ACTOR	1	
DESISTIMIENTO	1	
ADMISIÓN DE PRUEBA PERICIAL	1	

**Figura 4.5.5 Reporte estadístico**

Estos datos le son de gran ayuda al secretario auxiliar ya que puede realizar una mejor evaluación de su trabajo, viendo cuantos expedientes tiene pendientes y cuantos ya fueron terminados.

La ventana cuenta con un botón para mandar estos datos a un documento en Excel.  
Ver figura 4.5.6

Estadística31-07-2012(1) [Modo de compatibilidad] - Microsoft Excel

	A	B	C	D	E
		Descripción	Fecha	Fecha Hasta	
1					
2		RODRIGO ROBLES ENRÍQUEZ	01/01/201	31/07/201	
3					
4		Promociones de trámite turnadas	41		
5		Promociones de trámite pendientes de acuerdo	5		
6		Promociones de nuevo expediente turnadas	66		
7		Promociones de nuevo expediente pendientes de auto de presidencia	35		
8		Promociones devueltas	31		
9					
10		Total de acuerdos que desahogan promociones	59		
11		AVOCAMIENTO	2		
12		TRAMITE	1		
13		FALLO	1		
14		TURNO A PONENTE	1		
15		AUTO INICIAL	1		
16		COMPETENCIA DELEGADA	1		
17		RADICACIÓN	1		
18		CON SELLO	5		
19		AGREGADO	2		
20		PREVENCIÓN	1		
21		ACLARACIÓN DE DEMANDA	5		
22		ADMISIÓN DE LA DEMANDA	2		
23		DESECHAMIENTO DE DEMANDA	1		
24		CONTESTACIÓN DE DEMANDA	1		
25		ADMISIÓN DE LA AMPLIACIÓN DE DEMANDA	3		
26		PEDIMENTO PGR	1		
27		OPINIÓN DE PGR	1		
28		CONCEDE SUSPENSIÓN	1		
29		NIEGA SUSPENSIÓN	1		
30		DESAHOGA VISTA	1		
31		CUMPLE REQUERIMIENTO	1		
32		CONTESTA AMPLIACIÓN DE DEMANDA	1		

Figura 4.5.6 Datos estadísticos exportados a Excel

## CONCLUSIONES

La implementación del sistema de control y seguimiento de expedientes, permitirá hacer los procesos jurídicos más ágiles, rápidos y con mayor eficiencia, contribuyendo a la labor que se realiza en esta institución.

Se permitirá reducir significativamente los costos de cada proceso. El sistema abatirá los costos de copias, impresiones, carpetas y sobre todo el espacio en el que ahora están resguardados todos los expedientes que se llevan en la institución.

El sistema cumple con los requerimientos de la institución y se adecua a las necesidades de cada área, logrando así solventar las necesidades que se tienen y mejorar en varios aspectos, en beneficio de la productividad de la institución.

El desarrollo del sistema permitirá a la institución dar el primer paso hacia la automatización y optimización de los procesos, así como a la digitalización y modernización de los elementos que intervienen en ellos, elevando la confiabilidad de una institución tan importante en la ejecución de la ley en México.

Debido a que el número de usuarios que utilizan el sistema es muy grande fue necesario planear bien cómo restringir los permisos para que cada uno tenga acceso sólo a lo que necesita. Es por ello que la parte de los perfiles de usuario fue crucial en la planeación, facilitando al final la implementación de la seguridad.

Gracias a la metodología y a la programación en capas que se utilizó para su desarrollo, el sistema puede seguir creciendo de manera rápida y confiable. El hecho de que las funcionalidades principales como búsquedas y generación de reportes ya se encuentran desarrolladas ahorrará mucho tiempo a futuras aplicaciones.

Un aspecto importante del sistema fue el aspecto visual y el diseño de las ventanas. Los controles DevExpress tienen un efecto más llamativo que los controles normales

del Visual Studio, es por ello que se optó por esta herramienta. Los íconos fueron diseñados para que el usuario identifique la funcionalidad de los botones sin tener que pasar el puntero del mouse sobre éstos para que se muestre su descripción.

Las ventanas principales fueron diseñadas basadas en aplicaciones tipo Windows, con barras de herramientas principales como secundarias, menús superiores que clasifican las operaciones. Esto le ayuda al usuario a familiarizarse más rápido con el sistema y hacer un uso más eficiente.

## BIBLIOGRAFÍA

- Titulo: Análisis estructurado moderno  
Autor: Edward Yourdon  
Editorial: Prentice-hall hispanoamericana  
Primera edición 1993
- Titulo: Diseño y gestión de Sistemas de Bases de Datos  
Autor: Lucas Gómez Ángel  
Editorial: Paraninfo.
- Titulo: Introducción a los Sistemas de Bases de Datos, Volumen 1.  
Autor: Date C.J.  
Editorial: Addison-Wesley Iberoamericana
- Titulo: Ingeniería de software  
Autor: Ian Sommerville  
Editorial: Addison-Wesley Iberoamericana  
Año: 2002
- Titulo: Ingeniería del software  
Autor: Roger S. Pressman, Jose Maria Troya  
Editorial: McGraw Hill  
Año: 1988
- Titulo: Fundamentos de Bases de Datos  
Autor: Silberschatz, Abraham  
Editorial: McGraw-Hill, 2002
- Titulo: El Arte de probar el software (The Art of Software Testing) El Ateneo  
Autor: Glenford J. Myers

Editorial: John Wiley & Sons, Inc. 1979

Año: 1983

- Título: Aprendiendo UML en 24 horas  
Autor: Schmuller Joseph  
Editorial: Prentice Hall  
Año: 2005
- Título: Diseño Orientado a Objetos con UML  
Autor: Alarcón Raúl  
Editorial: Grupo Eidos  
Año: 2007
- Título: Manual de base de datos  
Autor: Figueroa Resendiz, Juana  
DGSCA Guías y textos de cómputo  
Año: Julio 2002
- <http://msdn.microsoft.com>