



Instituto Politécnico Nacional



**Sistema Generador de Documentos de Estructura Variable**



**No. de Reg. TT0308**

Nº de serie:

Serie: Amarilla  
Reporte Técnico

Mayo de 2002

***AUTORES:***

***De la Barrera Pliego Roberto Daniel***

***Sánchez Borjas Yessika***

***Soto Gómez Carol Fabiola***

***ASESOR:***

***Lic. Andrés Ortigoza Campos***

***RESUMEN:***

En este trabajo se describe el desarrollo y la puesta en marcha de un sistema de información automatizado, el sistema es capaz de diseñar, utilizar y generar documentos, que dependiendo del estado en el que se encuentre la información entregada, el documento pueda variar en su estructura y/o información.

### **ADVERTENCIA**

“Este informe contiene información desarrollada por la Escuela Superior de Cómputo perteneciente al Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convengan”.

La aplicación no convenida exime a la escuela de su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en la Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, sita en Avenida Juan de Dios Bátiz s/n, teléfono: 729-6000 extensión 52012

Índice:

<b>Temas</b>	<b>Págs.</b>
• Introducción -----	4
• Antecedentes	
✚ Estado del Arte -----	5
• Alcances -----	6
• CAPITULO 1	
✚ Fase de Análisis de SIGDEV -----	7
✚ 1.1 Definición del Sistema -----	7
✚- 1.1.2 Justificación del sistema-----	7
✚- 1.1.3 Objetivos -----	8
✚- 1.2 Plan de Proyecto-----	10
✚- 1.2.1Planificación de Desarrollo-----	10
✚- 1.2.2Herramientas y Técnicas Empleadas-----	10
✚- 1.2.2.1 Metodología para la Integración de Sistemas-----	11
✚- 1.2.2.2 Plataforma de Desarrollo -----	12
• CAPITULO 2 Fase de Diseño -----	14
✚ 2.1Descripción contextual -----	14
✚ 2.2 Diagrama de Flujo de Datos -----	15
✚ 2.3 Diccionario de datos de Proceso -----	18
✚ 2.4 Diccionario de Datos-----	25
✚ Modelo Lógico de la BD-----	26
✚ Relaciones entre tablas de la BD -----	27
✚ Diccionario de Datos de la BD-----	28
• CAPITULO 3 Descripción del Sistema-----	37
• CAPITULO 4 Marco Teórico-----	59
• MANUAL DE USUARIO-----	67
• MANUAL DE DISEÑADOR-----	
Bibliografía -----	74

## INTRODUCCIÓN

Actualmente para las organizaciones, resulta prioritario poder generar sus escritos de manera oportuna y correcta.

En las industrias la informática se ha convertido en un factor muy importante para el desempeño eficiente de sus actividades, ya que les resulta imprescindible llevar un adecuado seguimiento y control de sus documentos, lo cual se puede lograr si se cuenta con buenos mecanismos.

Las Notarías requieren de un área de sistemas que le proporcione herramientas de software eficientes para el procesamiento de sus altos volúmenes de información, que sean flexibles para responder adecuadamente a los requerimientos cambiantes de la misma, y con la infraestructura necesaria para proporcionar documentos de calidad con oportunidad reduciendo los tiempos de procesamiento.

Para satisfacer las necesidades de las Notarías se desarrolló el Sistema Generador de Documentos de Estructura Variable, el cual permite el registro, control y seguimiento de las operaciones realizadas por una Notaría.

Este sistema se desarrolló con ayuda de la Notaría No. 9. Al surgir dentro de la Notaría la necesidad de realizar documentos, se analizaron los requerimientos para poner en operación este tipo de aplicaciones, considerando principalmente la conveniencia y los requerimientos de las autoridades reguladoras.

Al surgir la necesidad de contar con un sistema que considerara todos los requerimientos para que fuera aceptado, se analizaron sistemas similares, pero no se adaptaban de forma total a sus necesidades además de ser costosos.

Para la mejor comprensión del desarrollo de dicho sistema, en este trabajo, primeramente se presentará un capítulo breve en el que se describe a grandes rasgos un marco de referencia que se necesitó como base para el entendimiento de los requerimientos del sistema, los siguientes tres capítulos muestran el desarrollo en sí del sistema en sus diferentes fases, la selección específica de herramientas tanto para análisis como para desarrollo y la documentación.

## Antecedentes

En la actualidad existen muchos sistemas de bases de datos, que relacionan el contenido de sus tablas con una plantilla de texto, para poder imprimir documentos de formatos fijos pero con datos diferentes según la referencia que tenga con la base de datos.

Microsoft Works es uno de ellos, pues relaciona un documento de edición de texto, con atributos de una base de datos creada en este mismo paquete, pero no se pueden crear una relación entre los documentos que utilizan la base de datos para poder crear un tercer documento relacionado con las variables existentes.

Existe un sistema que crea documentos Notariales llamado **KEA**, creado en 1980 por el Ing. David Carrillo. Este sistema trabaja en red utilizando en el servidor un sistema operativo llamado VMS, en un servidor VAX y Basic como el lenguaje donde fue creado el sistema. Como la mayoría de los equipos son PC's, se creó una interfaz de usuario para DOS y de ésta forma poder utilizar el sistema.

Este sistema se dedica únicamente a documentos Notariales (escrituras, actas y documentos. A pesar de que el sistema se limita sólo a este tipo de documentos, no tiene todas las variantes posibles para crear un documento determinado, por lo que el usuario depende del alcance del modelo en el que se basa el documento y necesita hacer las modificaciones manualmente. Todos los modelos para crear los documentos, están diseñados por el programador.

En 1998, una empresa mexicana (OASIS) y otra empresa francesa (DAXIS) se unieron para crear un sistema llamado Master Windows, que al paso de los meses y por cuestión de licencias se convirtió en Master Open dedicado también a documentos Notariales.

Este sistema utiliza una red Microsoft. El sistema operativo del sistema dentro del servidor es Windows NT 4.0 y su Base de Datos es SQL Server 6.5. El sistema está creado en Power Builder y Visual Basic para su ejecución dentro del servidor y de sus terminales.

El sistema se instala en todas las terminales para poder tener un trabajo más veloz y sus usuarios tienen delimitados los accesos por el sistema. Este se limita a documentos Notariales y es capaz de relacionar documentos diferentes con respecto a las diferentes participaciones de las personas o inmuebles dentro de la escritura. Los documentos, las tablas y las relaciones entre ellos, son creados por los usuarios, pues el sistema tiene un lenguaje interno para realizar esto, verificarlos y ejecutarlos.

### **Alcances**

El sistema constará de una interfaz amigable para el usuario, que le permita la realización de documentos cuya fórmula ha sido previamente desarrollada; por medio de un asistente se hará la recopilación de información necesaria para la elaboración de dicho documento. El sistema utilizará una Base de Datos donde se almacenará toda la información necesaria para la elaboración de dichos documentos, la cual podrá ser accesada fácilmente. El correcto llenado del documento será verificado por un intérprete que al interpretar la fórmula indicará la ausencia o equivocación en los datos recopilados, al ser interpretado exitosamente el documento se visualizará en un procesador de texto.

Además el sistema contará con la posibilidad de poder desarrollar sus propias fórmulas según las especificaciones requeridas para el documento

#### Requisitos del Software

- Un Manejador de Base de Datos  
De los manejadores existentes actualmente se llevó un análisis según los requerimientos del software, la disponibilidad y accesibilidad en costos. Tomando esto a consideración se decidió por SQL versión 7, no cerrándonos a la posibilidad de considerar otro manejador durante el desarrollo del sistema.
- Un procesador de Texto  
Debido a la Necesidad de hacer el sistema accesible en su implantación se decidió por el procesador de texto Word.
- Sistema Operativo  
Tomando en cuenta que el sistema va a ser comercial y compatible con el procesador de texto y el manejador de Base de datos se decidió por el sistema operativo Windows NT.

#### ❖ Restricciones del Sistema

- Sólo es para la elaboración concerniente a documentos Notariales.
- La computadora donde se instale el sistema necesitará tener instalado Word de Microsoft.
- Sistema Operativo Windows.

## CAPÍTULO 1 FASE DE ANÁLISIS DE SIGDEV

Para entender las necesidades de un sistema de información para el desarrollo de los diversos documentos que se desarrollan en una Notaría, se recomienda hacer una descripción preliminar de las características y naturaleza de las operaciones realizadas en dicho lugar.

### 1.1 DEFINICIÓN DEL SISTEMA

Trabajo Terminal: Sistema Generador de Documentos de Estructura Variable

Sitio de Investigación: Notaría No. 9

Ubicación:

#### 1.1.1 DEFINICIÓN DEL PROBLEMA

En las Notarías debido al servicio que dan, es de suma importancia la rapidez, eficiencia y calidad en el servicio que brindan a sus clientes.

Al realizar la investigación en dicha Notaría contaba con un sistema muy poco amigable y flexible lo que dificultaba la labor de la misma.

#### 1.1.2 JUSTIFICACIÓN DEL SISTEMA

Con el fin de apoyar a las funciones de operación de una Notaría para que sea más eficiente en la atención al cliente y en el manejo de la información para la toma de decisiones; se realizó la propuesta del siguiente sistema.

**SIGDEV** (Sistema de Generación de Documentos de Estructura Variable) el cual permitiría un desarrollo más adecuado y de una manera fácil a la Notaría, así como optimizar sus operaciones proporcionándole integridad y control en cada una de ellas. Además de contar con un protocolo de seguridad que garantiza la confidencialidad y seguridad de la información.

#### **Características del SIGDEV.**

##### **Amigable y Orientada al Usuario**

Para su operación no se requieren conocimientos especializados de computación. El sistema proporciona menús fáciles de entender,

 **Ventajas del Sistema**

- El sistema permite la elaboración de fórmulas por parte del Administrador ajustándolo a las necesidades cambiantes en este tipo de institución.
- Permite el diseño de Tablas según las necesidades de la Notaría.
- Genera dinámicamente las formas que se requieren para el llenado de dichas tablas.



### 1.1.3 OBJETIVOS

#### **Objetivo General:**

Se desarrollará un sistema que deberá ser capaz de diseñar, utilizar y generar documentos, que dependiendo del estado en el que se encuentre la información entregada, el documento pueda variar en su estructura y/o información.

#### **Objetivos particulares:**

- ✚ El sistema constará de una interfaz amigable para el usuario, que le permita la realización de documentos cuya fórmula ha sido previamente desarrollada; por medio de un asistente se hará la recopilación de información necesaria para la elaboración de dicho documento.
- ✚ El sistema utilizará una Base de Datos donde se almacenará toda la información necesaria para la elaboración de dichos documentos, la cual podrá ser manipulada fácilmente.
- ✚ El correcto llenado del documento será verificado por un intérprete que al interpretar la fórmula indicará la ausencia o equivocación en los datos recopilados, al ser interpretado exitosamente el documento se visualizará en un procesador de texto.
- ✚ El sistema contará con la posibilidad de poder desarrollar fórmulas propias de acuerdo con las especificaciones requeridas por el documento

## **1.2 PLAN DE PROYECTO**

La metodología del trabajo propuesta, consistió de las siguientes etapas:

1. -Construcción de prototipos funcionales para cada uno de los componentes del sistema para evitar desviaciones y errores en el corto plazo, y sobre todo asegurar la aceptación de los resultados parciales por parte de las áreas usuarias.
2. Asimilación de funcionalidad probada para posteriores prototipos .Adoptando los estándares de operación de la Notaría, lo anterior, disminuirá los problemas de aceptación de los prototipos y contrarresta la falta de tiempo por parte de los usuarios involucrados.
3. Establecimiento de mecanismos de evaluación de avances. Los cuales asegurarían de antemano el rápido establecimiento o restablecimiento de tareas que requerían de una mayor atención para su consecución o la reevaluación de prioridad.
4. Utilización de herramientas modernas de desarrollo. Las cuales facilitarían la construcción y modificación de los prototipos, así como, amortiguarían la curva de aprendizaje de la operación del sistema una vez concluido.
5. Establecimiento de estándares de diseño y programación. Los cuales facilitarían la comprensión de los modelos adoptados y, ante todo, el mantenimiento futuro de las aplicaciones del sistema.
6. Estandarización de la documentación técnica. Asegurando la existencia de la documentación básica de los sistemas al finalizar su desarrollo.

### **1.2.1 PLANIFICACIÓN DE DESARROLLO**

La objetiva determinación del alcance del proyecto, requería de una planeación adecuada de tareas a realizarse y de los tiempos que debían asociarse a ellas, debiendo quedar los objetivos a alcanzar y las metas a realizarse claramente documentados y adaptados por las áreas involucradas.

Mediante lo anterior, se propuso la realización del bosquejo preciso de las tareas que debían ser abordadas, lo que facilitaría el establecimiento de tiempos y recursos necesarios para la consecución del proyecto.

En esta fase se establecieron los siguientes puntos:

1. Procesos necesarios que debían ser automatizados.
2. Establecer la prioridad de los procesos.
3. Designar responsabilidades y responsables.
4. Determinar tiempos y secuencias de los procesos.

### **1.2.2 HERRAMIENTAS Y TÉCNICAS EMPLEADAS**

Para el desarrollo del proyecto nos auxiliamos de varias herramientas como son:

1. Racional Rose.
2. Power Designer.
3. Visio
4. C++ Builder 5
5. SQL Server

Y de Metodología MIS las cuales son descritas con mayor detalle a continuación.

### **1.2.2.1 METODOLOGÍA PARA LA INTEGRACIÓN DE SISTEMAS.**

Se apoya de conocimientos de diversas áreas que permiten lograr un mejor resultado de un proyecto de sistemas, estas áreas no sólo incluyen a la informática, sino también a la administración de proyectos, herramientas de comunicación gráfica y desarrollo de instalaciones. Saca provecho de las técnicas y métodos que permitan estar al día en lo que se refiere al control y administración de proyectos.

#### **FASES:**

- ✚ Fase de Análisis
- ✚ Fase de Infraestructura
- ✚ Fase de Diseño
- ✚ Fase de Construcción
- ✚ Fase de Implantación §

### **1.2.2.2. PLATAFORMA DE DESARROLLO**

Toda vez que se había manifestado el interés por parte de la Notaría No.9, de incursionar en el esquema de desarrollo cliente/servidor, que lo mantendría a la vanguardia tecnológica en cuanto a sistemas de información se refiere, se propuso la utilización de las siguientes herramientas para la construcción del sistema:

1. Herramienta case POWER DESIGNER, para la fase de diseño en el ámbito de conformación de la base de datos.
2. Herramienta VISIO, para la fase de diseño en el ámbito de conformación de los diagramas de flujo de datos.
3. Manejador de Base de Datos relacionales SQL SERVER 7.
4. Herramienta de Desarrollo C++ Builder 5.

Esta propuesta tenía como fundamento principalmente, cuatro aspectos importantes a mencionar.

1. La filosofía del sistema de la Notaría.  
Es decir, debería considerar el desarrollo del sistema integral de Notarías.
2. La Plataforma Cliente /Servidor  
El desarrollo de aplicaciones bajo este esquema, permite la mejor explotación de las ventajas que dan las PC's, las cuales sólo requerirán del software necesario para funcionar, como un esquema Cliente/Servidor. Considerando que se cuenta ya con

---

§ Kendall & Kendall. "Ingeniería de Software".

el hardware y el software requerido y con esto lograr la implantación exitosa del programa realizado.

3. Las Herramientas “CASE” de modelado.

Estas herramientas de modelado, denominadas “CASE”, facilitan el análisis y diseño para el desarrollo rápido de aplicaciones. Estas herramientas permiten de manera muy sencilla, e intuitiva, la construcción de Diagramas Esenciales o de Contexto, Diagramas de Flujo de Datos, Diagramas Entidad Relación y Diccionario de Datos; cada uno de los cuales, tiene una finalidad específica en las etapas de análisis y diseño de la aplicación, que se mencionarán adelante.

### **Herramientas de Modelado**

Dentro de las actividades que se recomiendan, en el desarrollo de los módulos de la aplicación se deben incorporar elementos como el análisis y diseño en un ciclo iterativo, es decir con un enfoque que pone a las actividades de análisis, diseño y construcción en un ciclo repetitivo para un número preestablecido de iteraciones, para cada uno de los módulos de la aplicación. Como herramientas, no sólo están contempladas las de desarrollo, también se incluyen las herramientas para el análisis y diseño, o modelado.

Algunas características de estas herramientas de modelado son:

- Gráficas con texto de apoyo.
- Permiten que el sistema sea visto en segmentos, de forma descendente (top-down).
- Reduce la redundancia(a la menor posible).
- Son transparentes al lector, es decir, el lector no tiene que detenerse a pensar que esta viendo una representación de un sistema en lugar de ver al sistema mismo.
- Ayudan a predecir el comportamiento del mismo.

### **Herramientas para el Análisis**

El propósito principal de la actividad de análisis es transformar sus entradas o factores principales, las políticas de usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagramas de entidad – relación y otras herramientas. La actividad de análisis implica el desarrollo de un **modelo de contexto**, que representa una descripción formal de lo que el sistema debe de hacer, independientemente de la naturaleza de la tecnología que se use para cubrir los requerimientos.

Las herramientas gráficas de modelado más importantes del análisis estructurado son los diagramas de flujo de datos (DFD), los diagramas entidad – relación (DER); y los diccionarios de datos, los cuales se describen a continuación.

### **Diagramas de Flujo de Datos (DFD).**

El diagrama de flujo de datos es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por “conductos” y “tanques” de “almacenamiento “. Debe de tenerse en mente que el DFD es tan sólo una de las

herramientas de modelado disponibles y que únicamente proporciona un punto de vista de un sistema, el orientado a las funciones.

Los componentes de un DFD, son el proceso, el flujo, el almacén y el terminador.<sup>§</sup>

## Diagrama a bloques

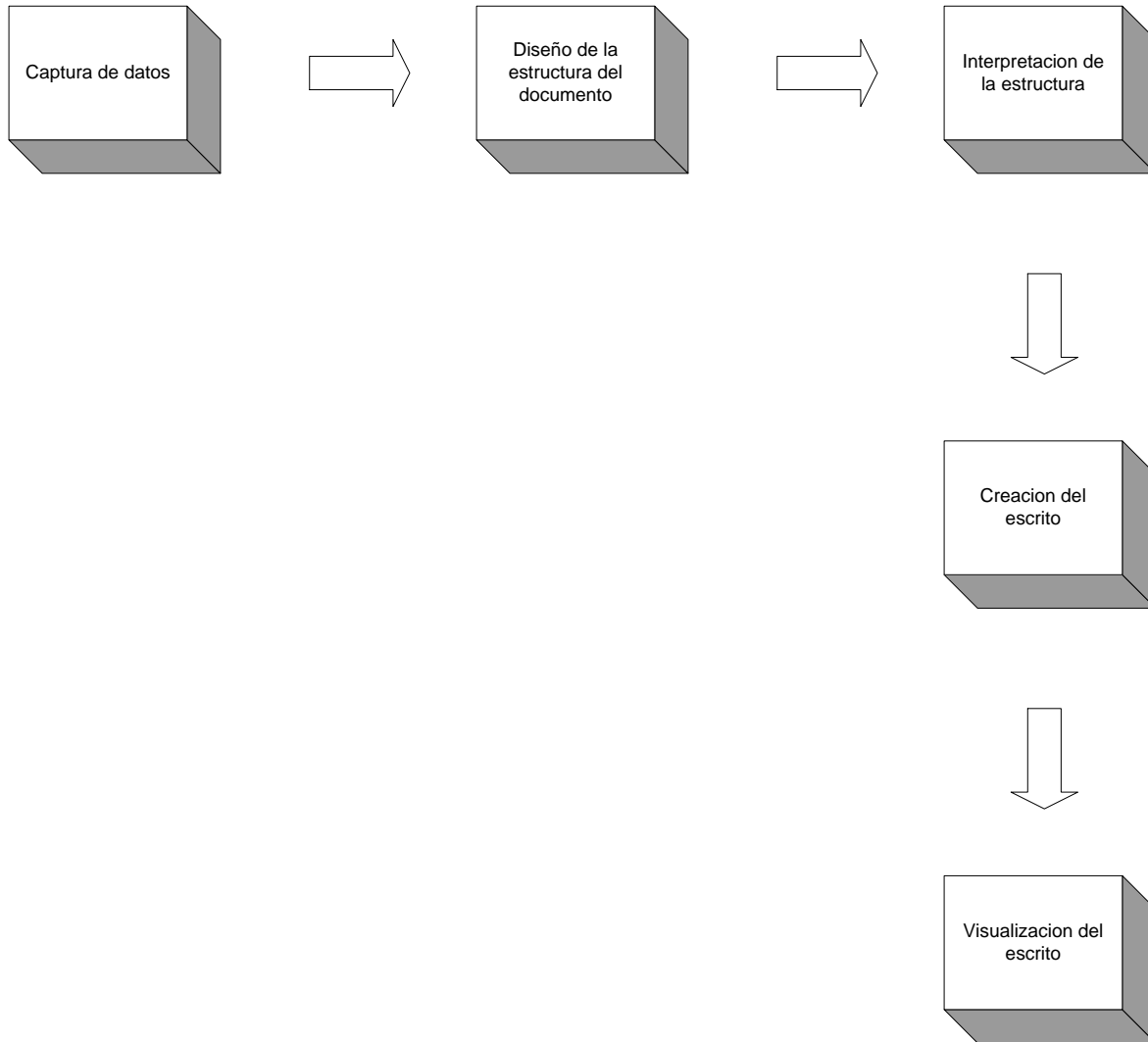


Figura 1. Diagrama a bloques

<sup>§</sup> Gane & Sarson. "Análisis estructurado de sistemas".

## CAPITULO 2 FASE DE DISEÑO

Como primera fase del análisis del sistema se realizó una investigación preliminar apoyada en diversas encuestas realizadas a las personas que se encontraban directamente relacionadas con el sistema, identificando a los usuarios responsables, así como las deficiencias actuales en el ambiente del usuario, y creando un campo de actividad inicial del sistema.

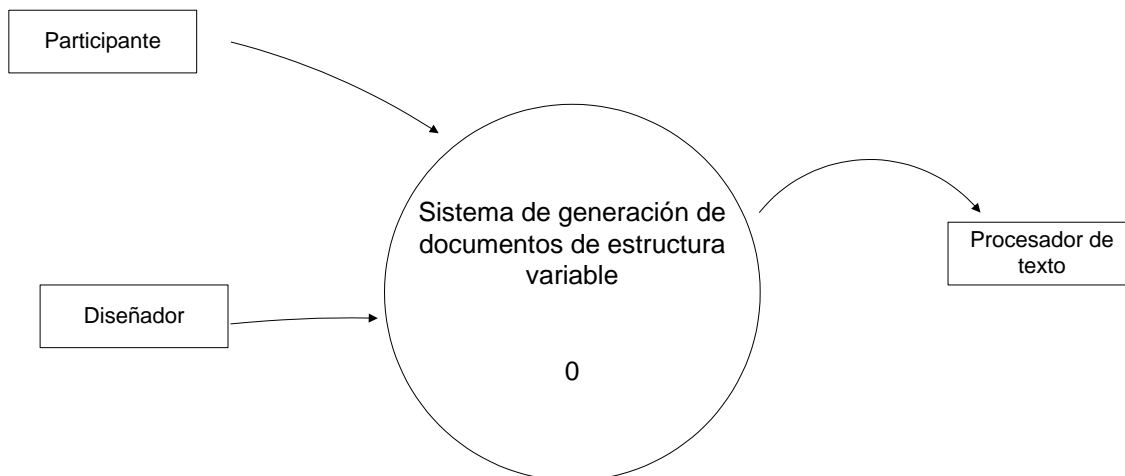
Dicha investigación permitió realizar el estudio de factibilidad, establecer metas y objetivos para el sistema a desarrollar y preparar el esquema que se utilizó como guía para el proyecto; iniciando así, el proceso de análisis del sistema. Cuyos resultados se plasman gráficamente para modelar el ambiente del usuario, representando una descripción formal de lo que el sistema debe hacer, independientemente de la naturaleza de la tecnología que se utilizará para cubrir los requerimientos.

### 2.1 DESCRIPCIÓN CONTEXTUAL

Antes de iniciar un análisis de los procesos más importantes que involucran la operación de SIGDEV. Se recomienda una descripción contextual dentro de la cual se desenvolverá el sistema, el diagrama se muestra la relación entre el sistema SIGDEV y las áreas que intervienen en el sistema integral.

Como se aprecia en el diagrama, el sistema SIGDEV aparece como un todo, el cual tiene iteración con las áreas de la empresa.

### Diagrama de Contexto



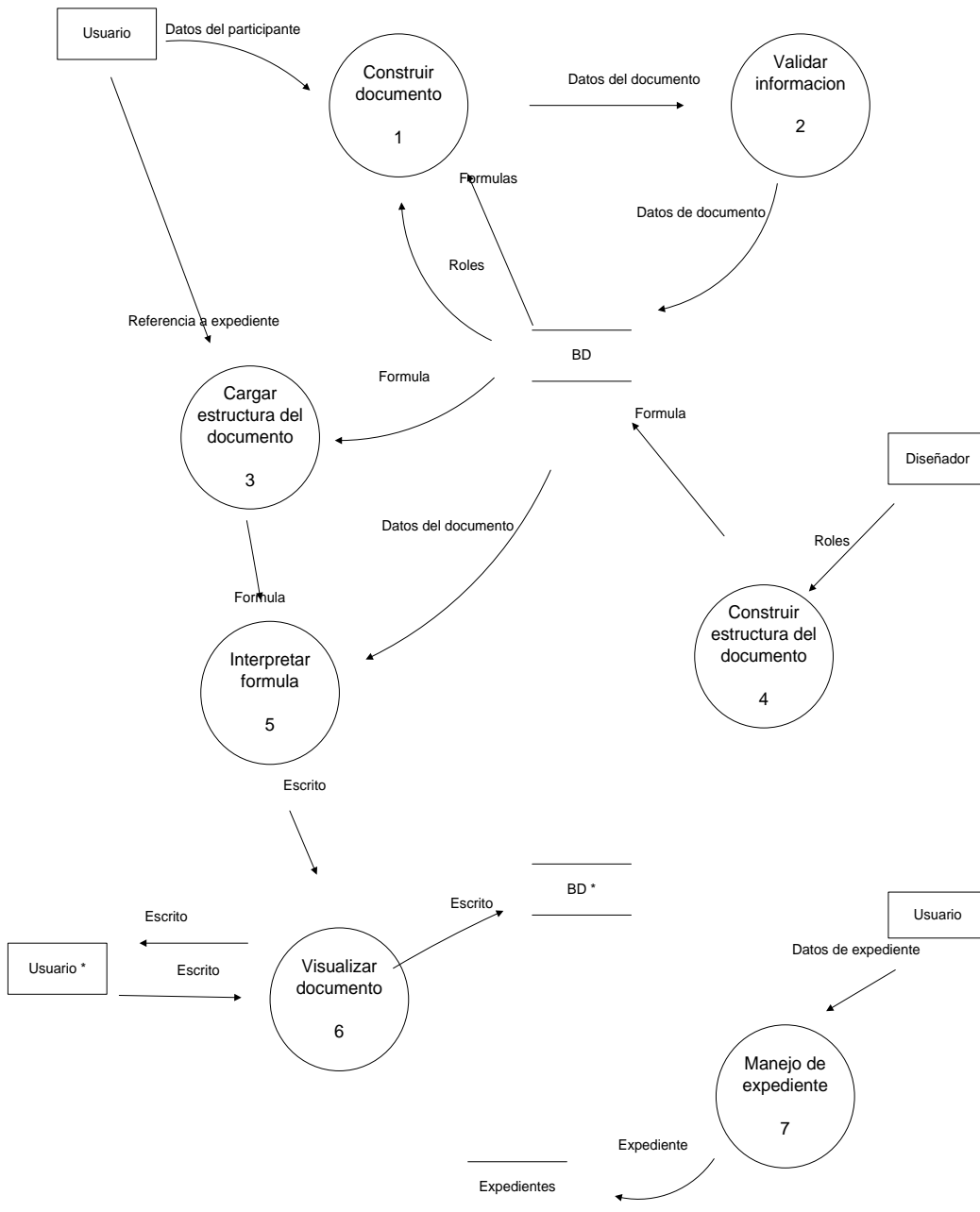
*Figura2.Diagrama de contexto nivel 0*

Analizando cada una de las entidades del diagrama, que representan a las áreas involucradas, se podrán establecer las entradas y salidas de información que debe tener el sistema para comunicarse con las áreas que así lo requieran, para determinar correctamente esta relación del sistema con las diferentes áreas se depende básicamente de encuestas realizadas con las personas que interactúan en el sistema.

## **2.2 DIAGRAMA DE FLUJO DE DATOS**

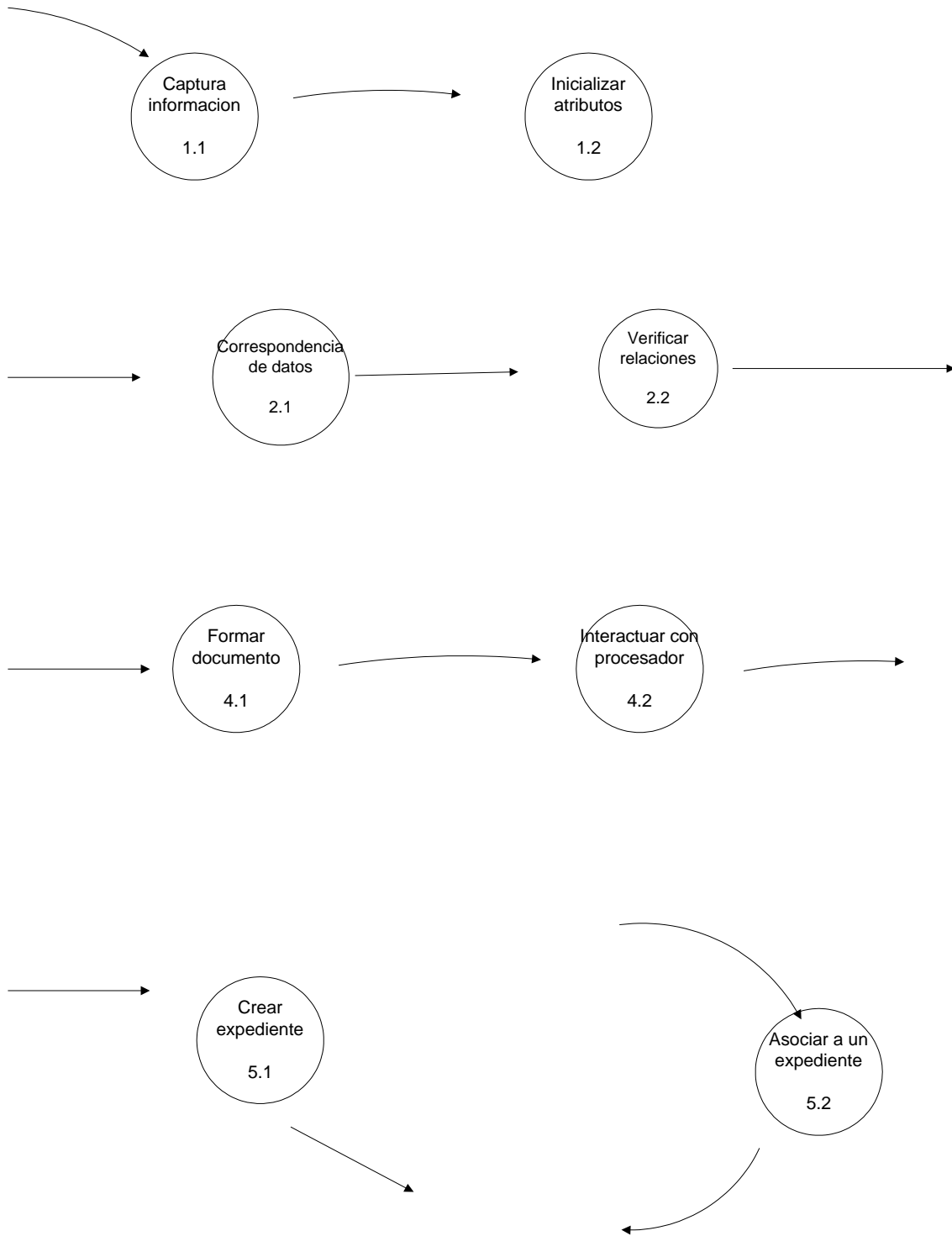
Se identificarán varios procesos funcionales, los cuales agrupan los procesos principales que engloban todas las operaciones. Estos procesos reciben y dejan información desde y hacia otros procesos por las áreas con las que se interrelaciona el sistema.

### Diagrama de Flujo Nivel 1



*Figura 3. Diagrama de contexto nivel 1*





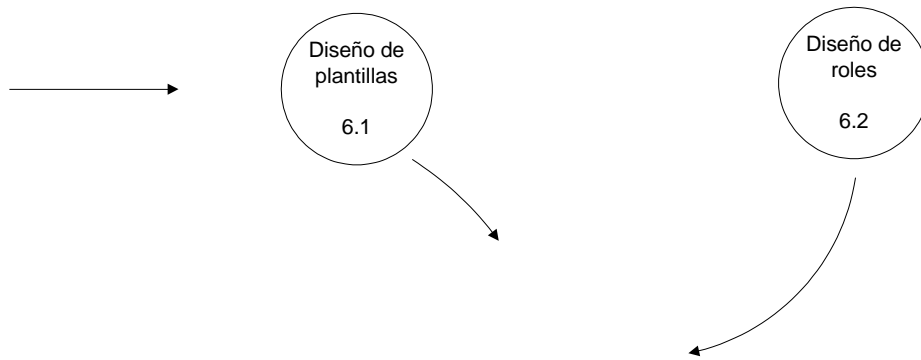


Figura 3. Diagramas de contexto nivel 2

**2.3 DICCIONARIO DE DATOS DE PROCESOS §**

Captura de Información

Proceso rel.1.1

Descripción: Se recopila Información necesaria de los participantes en el documento (personas, inmuebles, etc) para la elaboración de los mismos.

Entrada	Resumen de lógica	Salidas
Datos de participantes	Se recopila toda la información necesaria para la elaboración del documento seleccionado, de los participantes según su rol en el documento.	Datos del participante

Inicializar atributos

Proceso rel.1.2

Descripción: Se le asignan valores a los atributos del documento.

Entrada	Resumen de lógica	Salidas
Roles Datos de participante Fórmula	A cada participante se le asigna un rol (es) dentro del documento y al documento se le asigna una fórmula cual describe su estructura.	Documento

§ Kendal & Kendall. Ingeniería de Software.

Correspondencia de datos

Proceso rel.2.1

Descripción: Verifica que los datos de participantes otorgados por el usuario correspondan a la fórmula

Asignada al documento.

Entrada	Resumen de lógica	Salidas
Documento	La fórmula asignada al documento requiere ciertos participantes, la correspondencia de datos verifica que los datos asignados para el documento corresponda con los que requiere.	Documento

Verificar relaciones

Proceso rel.2.2

Descripción: verifica las relaciones que existen entre roles y participantes.

Entrada	Resumen de lógica	Salidas
Documento	Verifica que el rol(es) asignados a cada participante cumplan con las restricciones de cada rol.	Documento

Referenciar a datos de Expediente

Proceso rel.3.1

Descripción: Le indica al intérprete de la fórmula de que expediente debe de tomar los datos para crear el documento.

Entrada	Resumen de lógica	Salidas
Referencia a expediente Fórmula	Cuando se quiere crear a un documento se Tiene que cargar los datos del documento, por lo tanto se debe saber en donde están almacenados dichos datos, es decir en que expediente.	

Cargar fórmula

Proceso rel.3.2

Descripción: visualiza en el editor de texto la fórmula

Entrada	Resumen de lógica	Salidas
Referencia a expediente Fórmula	Cuando se va a crear un escrito se visualiza la fórmula en el editor de fórmulas para su posterior ejecución.	Fórmula

Sistema Generador de Documentos de Estructura Variable

Diseño de plantillas

Proceso rel.4.1

Descripción: Se crean las plantillas que se utilizarán en la creación de las fórmulas

Entrada	Resumen de lógica	Salidas
Datos plantilla	En el diseño de plantillas se crean éstas y se almacena el texto asociado a éste.	Plantillas

Creación de Fórmula

Proceso rel.4.3

Descripción: Se crea la fórmula del tipo de documento reuniendo plantillas

Entrada	Resumen de lógica	Salidas
Roles Plantillas	Se estructura la fórmula definiendo qué plantillas debe incluir y qué roles van a participar, intercalándolos según lo que se quiera.	Fórmula

Analizador sintáctico

Proceso rel.5.2

Descripción: Lleva a cabo el análisis sintáctico de la fórmula

Entrada	Resumen de lógica	Salidas
Componentes léxicos	Obtiene una cadena de componentes léxicos y comprueba si la cadena puede ser generada por la gramática establecida.	Errores

Manejo de errores

Proceso rel.5.3

Descripción: Los errores generados serán desplegados al usuario

Entrada	Resumen de lógica	Salidas
Errores	En la etapa de análisis sintáctico informará de cualquier error de sintaxis de manera inteligible este módulo se encargará de desplegarlos al usuario.	Mensajes durante interpretación
	para hacer su análisis.	

Formar documento

Proceso rel 6.1

Desplegar en el procesador de texto

Proceso rel.6.2

Descripción: Establece una colaboración con el procesador de texto.

Entrada	Resumen de lógica	Salidas
Escrito	Establece una comunicación con el procesador para que el sistema lo utilice como un medio para que el usuario visualice como quedará el documento generado.	Escrito

Crear expediente

Proceso rel.7.1

Descripción: Se genera un expediente

Entrada	Resumen de lógica	Salidas
Datos expediente Referencia al escrito	Al generar un documento se tiene que asociar a un expediente para su control.	Expediente



Asociar a un expediente

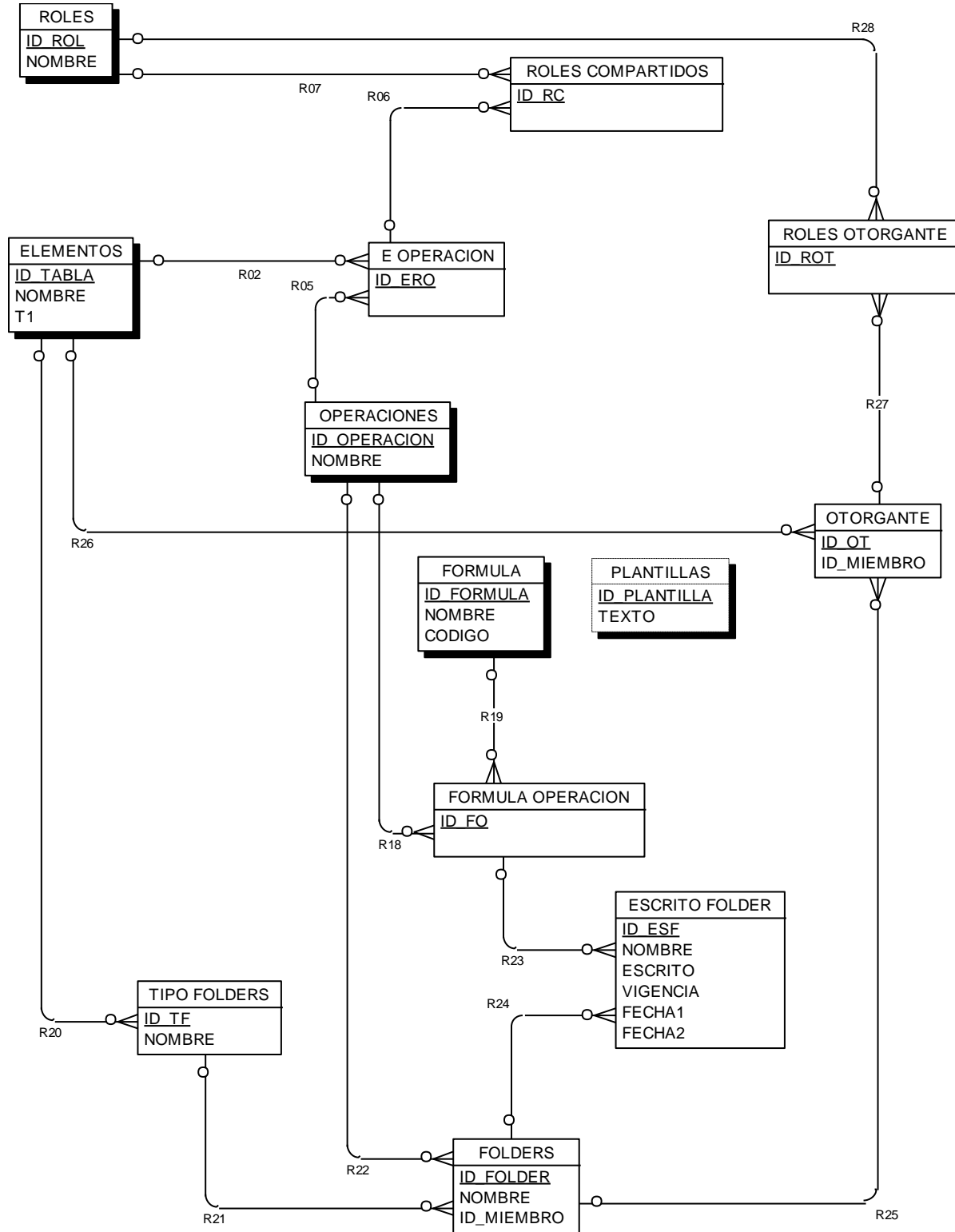
Proceso rel.7.2

Descripción: Asocia un documento a un expediente ya creado

---

Entrada	Resumen de lógica	Salidas
Referencia al escrito	Asocia el documento creado a un expediente ya existente para su control.	Expediente

**Modelo lógico de la base de datos**

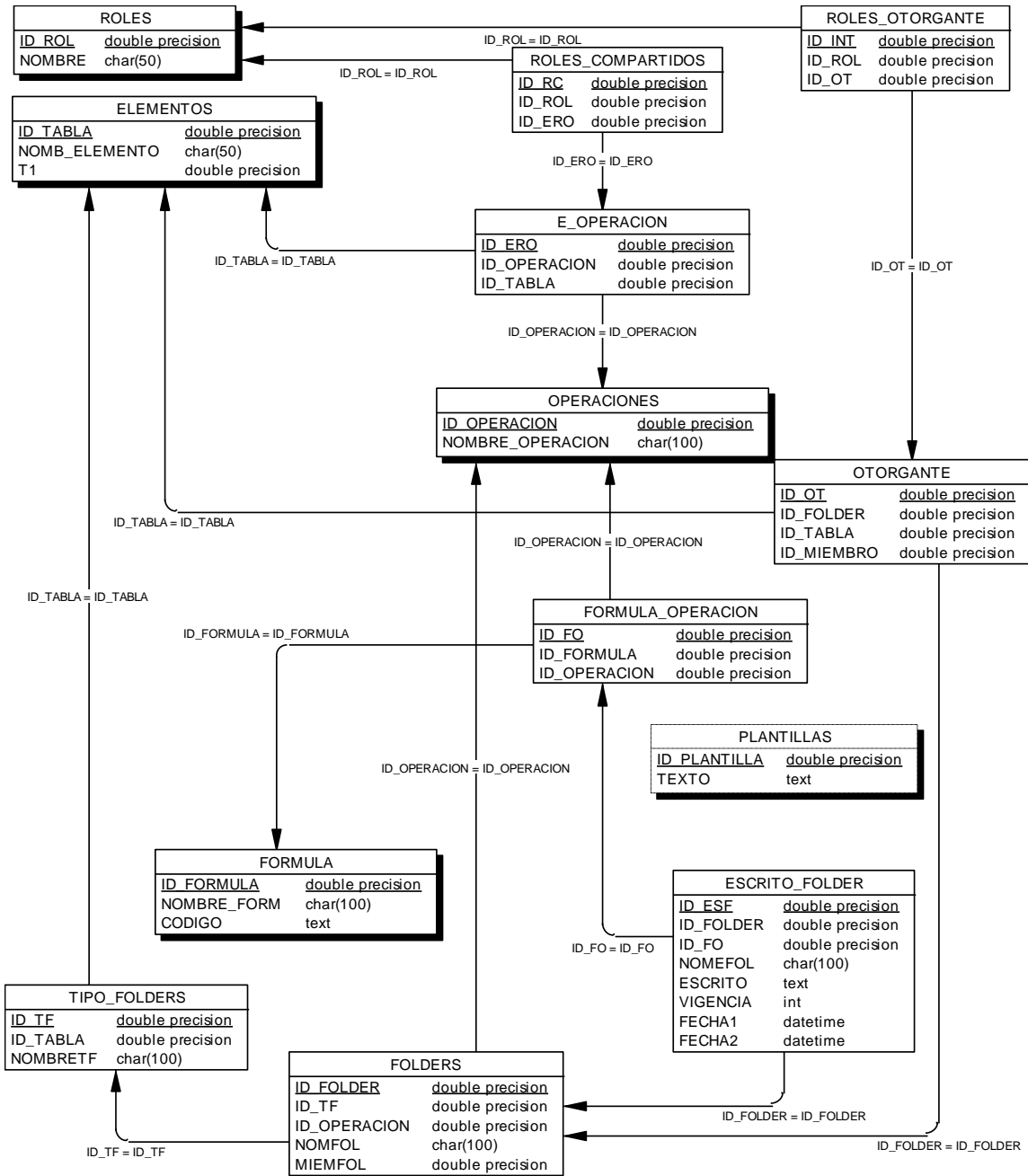


*Figura 4. Modelo lógico.*

**RELACIONES ENTRE TABLAS DE LA BASE DE DATOS**

<b>RELACION</b>	<b>DE LA TABLA</b>	<b>A LA TABLA</b>	<b>CARDINALIDAD</b>
R07	ROLES	ROLES_COMPARTIDOS	1..N
R28	ROLES	ROLES_OTORGANTE	1..N
R02	ELEMENTOS	E_OPERACIÓN	1..N
R26	ELEMENTOS	OTORGANTE	1..N
R20	ELEMENTOS	TIPO_FOLDERS	1..N
R06	E_OPERACIÓN	ROLES_COMPARTIDOS	1..N
R21	TIPO_FOLDERS	FOLDERS	1..N
R05	OPERACIONES	E_OPERACIÓN	1..N
R18	OPERACIONES	FORMULA_OPERACIÓN	1..N
R22	OPERACIONES	FOLDERS	1..N
R19	FORMULA	FORMULA_OPERACIÓN	1..N
R23	FORMULA_OPERACIÓN	ESCRITO_FOLDER	1..N
R24	FOLDERS	ESCRITO_FOLDER	1..N
R25	FOLDERS	OTORGANTE	1..N
R27	OTORGANTE	ROLES_OTORGANTE	1..N

**MODELO FÍSICO DE LA BASE DE DATOS**



*Figura 5. Modelo físico*

## 2.4 DICCIONARIO DE DATOS

En base a los diagramas, se realiza el diccionario de datos, el cual define de forma más precisa y de manera tal que tanto el usuario como el analista tengan entendimiento común de la información que se tiene almacenada en la base de datos, para poder explotarla de la forma más adecuada los datos que se representan en dicho diccionario son los siguientes.

### TABLAS

Nombre	Nombre Físico
<b>E OPERACIÓN</b>	<b>E OPERACIÓN</b>
<b>ELEMENTOS</b>	<b>ELEMENTOS</b>
<b>ESCRITO FOLDER</b>	<b>ESCRITO FOLDER</b>
<b>FOLDERS</b>	<b>FOLDERS</b>
<b>FORMULA</b>	<b>FORMULA</b>
<b>FORMULA OPERACIÓN</b>	<b>FORMULA OPERACIÓN</b>
<b>OPERACIONES</b>	<b>OPERACIONES</b>
<b>OTORGANTE</b>	<b>OTORGANTE</b>
<b>PLANTILLAS</b>	<b>PLANTILLAS</b>
<b>ROLES</b>	<b>ROLES</b>
<b>ROLES COMPARTIDOS</b>	<b>ROLES COMPARTIDOS</b>
<b>ROLES OTORGANTE</b>	<b>ROLES OTORGANTE</b>
<b>TIPO FOLDERS</b>	<b>TIPO FOLDERS</b>

**Tabla E OPERACION**

<p><b>Nombre:</b> E OPERACION</p> <p><b>Nombre Físico:</b> E _ OPERACIÓN</p> <p><b>Descripción:</b> Nos indica que TABLAS (elementos) se pueden utilizar en una OPERACIÓN determinado.</p>
--

### LISTA DE ATRIBUTOS

Nombre	Tipo	1	M
ID ERO	Float	Si	Si

### Descripción de ID\_ERO:

Nos ayuda a identificar un elemento con un rol determinado dentro de un trámite.

### REFERENCIA CON LAS SIGUIENTES TABAS:

TABLA	Cardinalidad	Dep.	Relacionado con:
ELEMENTOS	0,1	No	R02(R02)
OPERACIONES	0,1	No	R05(R05)
ROLES COMPARTIDOS	0,n	No	R06(R06)

**Tabla ELEMENTOS**

<b>Nombre:</b> ELEMENTOS
<b>Nombre Físico:</b> ELEMENTOS
<b>Descripción:</b> ELEMENTOS: contiene las referencias a las Tablas creadas para utilizarlas en las operaciones. Existen TABLAS que únicamente, son la referencia a otras tablas, a las cuales se les denomina TABLAS COMPARTIDAS.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	1	M
ID TABLA	ID_TABLA	Float	Si	Si
NOMBRE	NOMBRE	Char(50)	No	Si
TI	TI	Float	No	Si

**Descripción de ID\_TABLA:**

Nos ayuda a identificar la tabla que representa a algún elemento del documento.

**Descripción de NOMBRE:**

Es el nombre de la tabla a la que nos estamos refiriendo.

**Descripción de TI:**

Si las tablas es el resultado de la unión de dos tablas, colocamos el ID de la 1er tabla en este atributo.

**REFERENCIA CON LAS SIGUIENTES TABLAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
E OPERACION(E OPERACION)	0,n	No	R02(R02)
TIPO FOLDERS(TIPO FOLDERS)	0,n	No	R20(R20)
OTORGANTE(OTORGANTE)	0,n	No	R26(R26)

**Tabla ESCRITO FOLDER**

<b>Nombre:</b> ESCRITO FOLDER
<b>Nombre Físico:</b> ESCRITO_FOLDER
<b>Descripción:</b> Contiene los escritos realizados en un FOLDER de una OPERACION

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	1	M
<b>ID ESF</b>	ID ESF	Float	Si	Si
NOMBRE	NOMEFOL	char(100)	No	No
ESCRITO	ESCRITO	char(100)	No	No
VIGENCIA	VIGENCIA	Int.	No	No
FECHA 1	FECHA 1	Date	No	No
FECHA2	FECHA2	Date	No	No

**Descripción de ID\_ESF:**

Es el identificador del escrito, dentro de un Fólder.

**Descripción de NOMBRE:**

Es el nombre del Escrito.

**Descripción de ESCRITO:**

Archivo de texto, donde se almacena un documento. **Descripción de VIGENCIA:**

Cuánto tiempo tengo para que acepten mi documento.

**Descripción de FECHA1:**

Es la fecha de expedición del documento.

**Descripción de FECHA2:**

Fecha de aceptación.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
FORMULA	0,1	No	R23(R23)
OPERACION FOLDERS	0,1	No	R24(R24)

**Tabla FOLDERS**

**Nombre:** FOLDERS

**Nombre Físico:** FOLDERS

**Descripción;** Se conforma por un TIPO DE FOLDER y UNA OPERACION, y nos ayuda a indicar que una OPERACION está dentro de que TIPO DE FOLDER.

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	1	M
ID FOLDER	ID FOLDER	Float	Si	Si
NOMBRE	NOMBRE	char(100)	No	No
ID MIEMBRO	ID MIEMBRO	float	No	No

**Descripción de ID\_FOLDER:**

Nos ayuda a identificar FOLDER

**Descripción de NOMBRE:**

El nombre del folder.

**Descripción de ID\_MIEMBRO:**

El ID MIEMBRO de la tabla relacionada.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
TIPO FOLDERS	0,1	No	R21(R21)
OPERACIONES	0,1	No	R22(R22)
ESCRITO	0,n	No	R24(R24)
FOLDER OTORGANTE	0,n	No	R25(R25)

**Tabla FORMULA**

<b>Nombre:</b>	FORMULA
<b>Nombre Físico:</b>	FORMULA Contiene el CODIGO necesario para realizar un OPERACION
<b>Descripción:</b>	DOCUMENTO de una determinada.

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	1	M
ID FORMULA	ID FORMULA	Float	Si	Si
NOMBRE	NOMBRE	char(100)	No	No
CODIGO	<b>CODIGO</b>	MEMO	No	

**Descripción de ID\_FORMULA:**

Nos ayuda a identificar una fórmula

**Descripción de NOMBRE:**

Es el nombre de la fórmula.

**Descripción de CODIGO:**

Contiene la información para crear el documento.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
FORMULA OPERACIÓN	0,n	No	R19(R19)

**Tabla FORMULA OPERACION**

<b>Nombre:</b>	FORMULA OPERACION
<b>Nombre Físico:</b>	FÓRMULA _ OPERACIÓN
<b>Descripción:</b>	Mantiene la Relación entre las FORMULAS y las OPERACIONES.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	1	M
ID FO	ID FO	Float	Si	Si

**Descripción de ID\_FO:**

Nos ayuda a identificar una FORMULA dentro de una OPERACIÓN.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
OPERACIONES	0,1	No	R18(R18)
FORMULA	0,1	No	R19(R19)
ESCRITO FOLDER	0.n	No	R23(R23)



**Tabla OPERACIONES**

**Nombre:** OPERACIONES  
**Nombre Físico:** OPERACIONES  
**Descripción:**  
 OPERACIONES: son los indicadores de los documentos que podemos realizar.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	I	M
ID- OPERACION	ID OPERACION	float	Si	Si
NOMBRE	NOMBRE	char(100)	No	

**Descripción de ID OPERACION**

El identificador de la OPERACION

**Descripción de NOMBRE**

Es el nombre de una operación determinada

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
E OPERACION	0,n	No	R05(R05)
FORMULA	0,n	No	R18(R18)
OPERACION FOLDERS	0,n	No	R22(R22)

**Tabla OTORGANTE**

**Nombre:** OTORGANTE  
**Nombre Físico:** OTORGANTE  
**Descripción:** Contiene la relación de MIEMBROS de una TABLA con un FOLDER determinado.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	1	M
ID OT	ID OT	Float	Si	Si
ID MIEMBRO	ID MIEMBRO	Float	No	No

**Descripción de ID\_OT**

Nos ayuda a Identificar Otorgantes y los Roles relacionados con el.

**Descripción de ID\_MIEMBRO**

Es el ID de la tabla a la que se refiere nuestro documento.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relacionado con:
-------	--------------	------	------------------

<b>FOLDERS</b>	0,1	No	R25(R25)
ELEMENTOS	0,1	No	R26(R26)
ROLES OTORGANTE	0,n	No	R27(R27)

**Tabla PLANTILLAS**

**Nombre:** PLANTILLAS  
**Nombre Físico:** PLANTILLAS  
**Descripción:** Contiene fragmentos de Texto que ^c Ocupan para completar documentos.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	1	M
ID PLANTILLA	ID_PLANTILLA	float	Si	Si
NOMBRE	NOMBRE	char(255)	No	No
TEXTO	TEXTO	MEMO	No	No

**Descripción de ID\_PLANTILLA**

Identificador de Plantilla.

**Descripción de NOMBRE**

Nombre de Plantilla.

**Descripción de TEXTO**

Contiene el texto a insertar en los documento.

**Tabla ROLES**

**Nombre:** ROLES  
**Nombre Físico:** ROLES  
**Descripción:** ROLES: son todos los papeles que puede desempeñar un ELEMENTO determinado dentro de una OPERACION determinada.

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	1	M
<b>ID ROL NOMBRE</b>	ID ROL NOMBRE	float char(50)	Si No	Si

**Descripción de ID\_ROL**

Nos ayuda a identificar el rol dentro del sistema.

**Descripción de NOMBRE**

Es el nombre de la tabla a la que nos referimos

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalid	Dep.	Relación
ROLES COMPARTIDOS	0,n	No	R07(R07)
ROLES OTORGANTE	0,n	No	R28(R28)

**Tabla ROLES COMPARTIDOS**

**Nombre:** ROLES COMPARTIDOS  
**Nombre Físico:** ROLES\_COMPARTIDOS  
**Descripción:** Son los Roles que puede tener una TABLA determinada, en un Operación determinada.

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	I	M
<b>ID RC</b>	ID RC	float	Si	Si

**Descripción de ID\_RC**

Nos ayuda a identificar el rol que comparte un ELEMENTO con su ROL determinado.

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalid	Dep.	Relación
E OPERACION	0,1	No	R06(R06)
ROLES	0,1	No	R07(R07)

**Tabla ROLES OTORGANTE**

**Nombre:** ROLES OTORGANTE  
**Nombre Físico:** ROLES\_OTORGANTE  
**Descripción:** Son los ROLES que tiene un OTORGANTE en un FOLDER y OPERACIÓN determinada.

**LISTA DE ATRIBUTOS**

Nombre	Nombre Físico	Tipo	1	M
<b>ID ROT</b>	ID INT	float	Si	Si

**Descripción de ID\_ROT**

Identificador de Rol de Otorgante

**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalid	Dep.	Relación
OTORGANTE(OTORGANTE)	0,1	No	R27(R27) R28(R28)
ROLES(ROLES)	0,1	No	

**Tabla TIPO FOLDERS**

**Nombre:** TIPO FOLDERS  
**Nombre Físico:** TIPO\_FOLDERS  
**Descripción:** Está relacionada a una TABLA determinada, la cual nos indica las Características del FOLDER. EL fólдер, nos ayuda a definir cuáles son los Datos GENERALES, dentro de todos los DOCUMENTOS de una OPERACION determinada.

**Lista de Atributos**

Nombre	Nombre Físico	Tipo	1	M
<b>ID TF</b>	ID TF	float	Si	Si
NOMBRE	NOMBRETF	char(100)	No	No

**Descripción de ID\_TF:**

Identifica Tipos de

fólder **Descripción de**

**NOMBRE:**

El nombre del fólder.

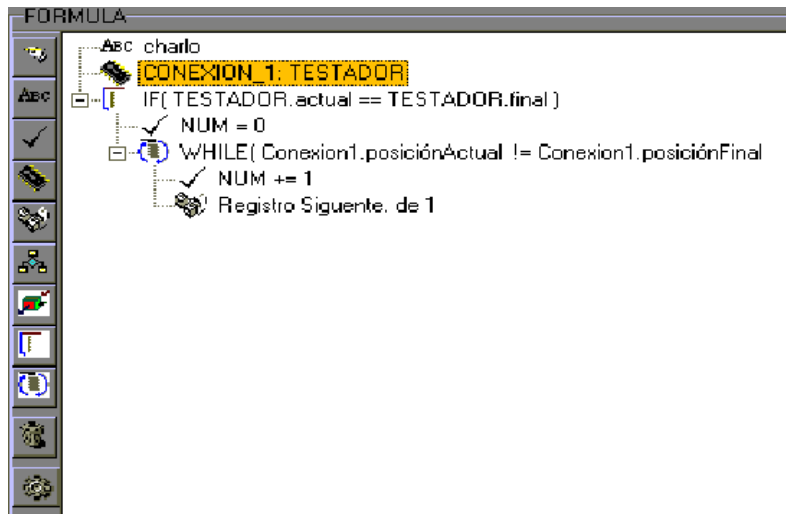
**REFERENCIA CON LAS SIGUIENTES TABAS:**

Tabla	Cardinalidad	Dep.	Relación
ELEMENTOS	0,1	No	R20(R20)
FOLDERS	0,n	No	R21(R21)

## CAPITULO 3 DESCRIPCIÓN DEL SISTEMA

### MODULO DE MANEJO DEL INTÉRPRETE

La gramática del intérprete, no tiene problemas de sintaxis, pues se utiliza un EDITOR DE FORMULA o Asistente, que inserta y/o modifica código en orden y sin errores. El usuario nunca ve el código en modo texto, pues el EDITOR DE FORMULA lo despliega en forma de árbol, indicando las características de cada nodo en forma visual.



*Figura 6. Interfaz del editor de fórmulas*

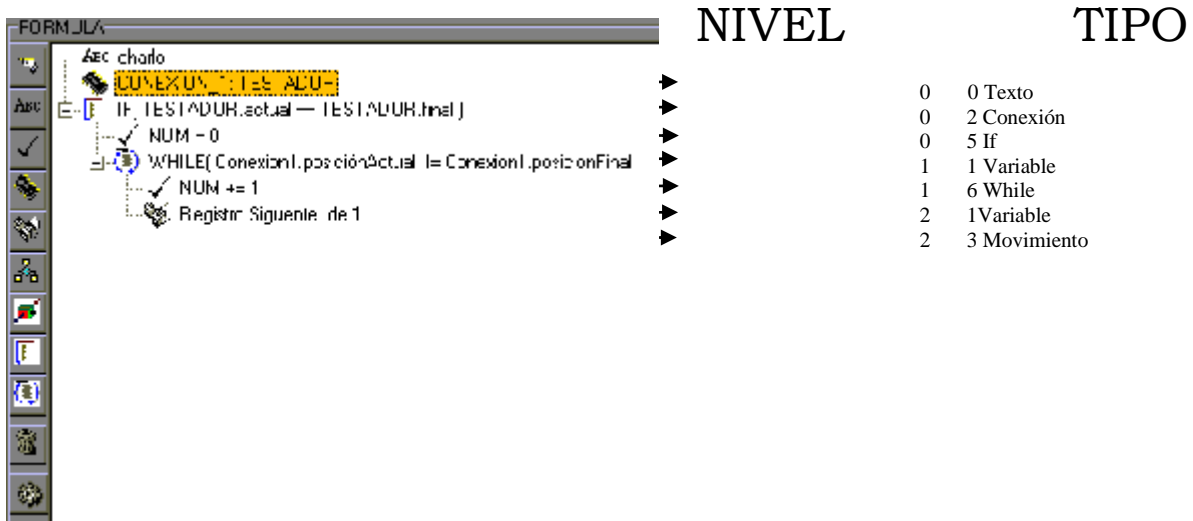
Para explicar la forma en que se Interpreta el código, tenemos que saber que dentro del código, después de una línea de texto existe un SALTO DE LINEA ( \n ), el cual determina un NODO y dentro de esa línea de código, existen TABULACIONES ( \t ) que nos ayudan a determinar las características de ese NODO en particular.

Todos los nodos tienen 3 características en común:

0	1	Texto: HOLA	.....	\n
0	2	CONEXIÓN 1: Testador	.....	\n
0	5	VAR1: Numero	.....	\n
1	6	IF(TESTADOR.SEXO==TRUE)	.....	\n
2	8	WHILE( Testador.RegistroActual j= Testador.RegistroFinal )	.....	\n

NIVEL	TIPO	Descripción del Contenido del Nodo.	...	SALTO
0	0	Texto: HOLA	..... ..	\n
0	1	CONEXIÓN 1: Testador	..... ..	\n
0	3	VAR1: Numero	..... .	\n
0	5	IF(TESTADOR.SEXO==TRUE)	..... ..	\n
1	6	WHILE( Testador.RegistroActual Testador.RegistroFinal )	i= ..... ..	\n

a) **NIVEL**, es muy importante, pues nos indica si el NODO SIGUIENTE pertenece a el nodo actual o si es sólo el SIGUIENTE NODO. Es fundamental, para saber qué elementos se encuentran dentro de la función IF o cuáles están dentro de la Función While.



*Figura 7.Descripción de los nodos y niveles*

- b) **TIPO**, existen 9 tipos de Nodo diferentes, los cuales tiene funciones diferentes para el intérprete y para la creación del documento.
- c) **DESCRIPCION**, es el texto que el usuario ve frente al NODO, el cual da una explicación general de lo que va a realizar el nodo.
- d) **...**, serán las características de ese nodo en particular.
- e) **\n**, es el salto de línea que indica el fin de las características de un NODO.

Los 9 tipos de Nodo que existen son los SIGUIENTES.

NOMBRE	IDENTIFICADOR DE TIPO	BOTON QUE LO CREA.
Comentario		
Texto		
Conexión		
Movimiento		
Variable		
Dato de la Base		
IF		
WHILE		
Función		

*Figura 8. Descripción de los nodos del editor.*

## Tipo “0” Texto

❖ *Objetivo:* Con este componente, se inserta un segmento de texto dentro del documento.

❖ *Atributos:*

1. Texto: Identificador relacionado con la Tabla Plantillas, donde está contenido el texto a guardar.
2. TiL: Tipo de letra.
3. TaL: *Tamaño de Letras.*
4. Caract: Son las características que tendrá el texto dentro del documento. Es un arreglo de 7 caracteres:

1	2	3	4	5	6	7
3	J	-	-	F	-	N

1. 0      Letras en Negritas.
- 1      Letras en Cursiva.
- 2      Letras en Negrita y Cursiva.

3	Letras tal cual como fueron escritas.
2. I	Alinear Izquierda.
D	Alinear Derecha.
C	Alinear al Centro.
J	Alinear Justificado.
3. T	Tabulación.
4. S	Salto de página.
5. F	Espacio en blanco al frente del texto.
6. A	Espacio en blanco detrás del texto.
7. N	Letras tal cual como fueron escritas.
M	minúsculas.
Y	mayúsculas.

❖ *Código a Leer:*

No.	A	B	C	D Texto	E TiL	F TaL	G Caract
1	0	0	Hola	1	Times New Roma	12	3J—F-N

- A) Indica el nivel donde se encuentra el nodo.
  - B) Indica el tipo de Nodo, que en este caso es 1.
  - C) El texto que ve el usuario como descripción del nodo.
  - D) El ID de la Tabla PLANTILLAS.
  - E) Tipo de letra a utilizar en el texto seleccionado.
  - F) El tamaño de letra del texto seleccionado.
- Las características de cómo se verá el texto en el documento.

❖ *Procedimiento:*

1. Se busca el ID de Platilla contenida en el Atributo Texto.
2. Se toma el texto correspondiente a este ID.
3. Se aplican las características marcadas en el Atributo Caract.
4. Se aplica el Tipo de Letra Contenido en el Atributo TiL.
5. Se aplica el Tamaño de Letra Contenido en el Atributo TaL.
6. Se inserta el Texto al Documento.

## Tipo “1” Conexión

- ❖ *Objetivo:* Este componente, realiza una consulta SQL, donde selecciona OTORGANTES con un ROL determinado de un FÓLDER con una OPERACIÓN determinada, para después manipular la información desde un Objeto de tipo **Movimiento(Tipo “2”) o DatoDeLaBase (Tipo “4”)**



Ejem. Seleccionamos lo OTORGANTES que tengan el ROL testador y el FÓLDER 1 que tiene la operación TESTAMENTO.

❖ *Atributos:*

1. ADO: El Intérprete puede crear 1, 000,000 de conexiones a la Base de Datos en una sola Fórmula. El ADO es un Identificador dentro del Arreglo de Conexiones a la Base de Datos.
2. Consulta: Tiene la Consulta SQL abreviada en 2 partes: Ejem.:
  - *T4R1*
    1. *T es el Tipo de Tabla que debe tener el otorgante, en este caso es la Tabla 4.*
    2. *R se el Rol al que debe estar asociado el Otorgante, en este caso es el Rol 1.*

No Contemplamos el FÓLDER o la OPERACIÓN, pues estos datos varían según donde esté interpretándose.

3. Nombre: Nos ayuda a que el usuario le identifique más fácilmente.

❖ *Código a Leer:*

No	A	B	C	D	E	F
1	0	1	Conexión1 Persona Física Testador.	1	T4R1	Persona Física Testador.
2	0	1	Conexión2 Persona Física Heredero.	2	T4R2	Persona Física Heredero.

- G) Indica el nivel donde se encuentra el nodo.
- H) Indica el tipo de Nodo, que en este caso es 1.
- I) El texto que ve el usuario como descripción del nodo.
- J) El identificador del Nodo dentro del Intérprete.
- K) La consulta a realizar.
- L) El nombre de la consulta, facilitada por el usuario.

❖ *Procedimiento:*

1. Nos colocamos en el arreglo de Conexiones del Intérprete, dependiendo del contenido de D.
2. Tomamos el ID\_FÓLDER (se le proporciona al intérprete cuando inicia la interpretación ), creamos una consulta SQL con las características de E y el ID\_FÓLDER.

3. Ejecutamos la consulta.

## Tipo “2” Movimiento

- ❖ *Objetivo:* Son los movimientos entre los registros de una conexión determinada.
  
- ❖ *Atributos:*
  1. ADO: El Intérpete puede crear 1,000,000 de conexiones a la Base de Datos en una sola Fórmula. El ADO es un Identificador dentro del Arreglo de Conexiones a la Base de Datos.
  2. Acción: Son los movimientos dentro de la conexión. Son 5 acciones diferentes:
    1. *Primer Registro.*
    2. *Registro Anterior.*
    3. *Registro Siguiete.*
    4. *Ultimo Registro.*
    5. *Fin de registros.*

- ❖ *Código a Leer:*

No.	A	B	C	D	E
1	0	2	Conexión1.PrimerRegistro	1	1
2	0	2	Conexión4.FinDeRegistro.	4	4

- Indica el nivel donde se encuentra el nodo.
  - B) Indica el tipo de Nodo, que en este caso es 1.
  - C) El texto que ve el usuario como descripción del nodo.
  - D) El ADO o Identificador de conexión con el que haremos el movimiento entre registros.
  - E) El movimiento a realizar dentro los registros.

- ❖ *Procedimiento:*

1. Nos colocamos en el Arreglo de conexiones según el contenido de D.
2. Realizamos un movimiento entre los registros dependiendo del valor que tenga E.

## Tipo “3” Variable

❖ *Objetivo:* Son variables numéricas utilizadas para realizar operaciones matemáticas, ser utilizadas como banderas o imprimirlas en el documento.

❖ *Atributos:*

1. Var1: En el intérprete existe un arreglo de variables. El contenido de Var1 es un identificado del arreglo.
2. Nombre1: El nombre de la Variable 1.
3. Signo: Es el signo de la operación a realizar entre variables o la operación de una variable y un número. (+, -, \*, = y /).
4. Var2: En el intérprete existe un arreglo de variables. El contenido de Var2 es un identificador del arreglo o el valor de un dato.
5. Nombre2: Es el nombre de la segunda variable.

❖ *Código a Leer:*

No	A	B	C	D Var1	E Nombre1	F Signo	G Var2	H Nombre2
1	0	3	Var1=5	1	Var1	=	5.33	
2	0	3	Var2*=Var1	2	Var2	*=	1	Var1

- A. Indica el nivel donde se encuentra el nodo.
- B. Indica el tipo de Nodo, que en este caso es 3.
- C. El texto que ve el usuario como descripción del nodo.
- D. El identificador de la Variable.
- E. El nombre de la Variable.
- F. El signo a aplicar entre Variables o Variable y número.
- G. El identificador de la segunda variable y el número con el que se aplica la operación.
- H. El nombre de la Variable.

❖ *Procedimiento:*

1. Nos colocamos en el arreglo de Variables con el ID contenido en D.
2. Verificamos si H tiene el nombre de otra variable. Si tiene el nombre de otra variable, buscamos la otra variable en el arreglo de Variables con el ID contenido en G, si H no tiene nombre, tomamos como dato lo contenido en G.

3. Dependiendo del signo que se encuentre en F se realiza la operación matemática y se guarda en la primer Variable.

## Tipo “4” DatoDeLaBase

❖ *Objetivo:* Obtener Datos de una conexión, para después insertarlos en el documento.

❖ *Atributos:*

1. ADO: Es el identificador del Arreglo de Conexiones del Intérprete al que nos referiremos.
2. Sub: En el sistema existen Tablas compuestas es decir, una tabla que hace referencia a otras tablas. Si este Atributo tiene datos, significa que la conexión utiliza una tabla compuesta y nos indica la posición en la que se encuentra la referencia de la Tabla que queremos consultar.
3. Atrib: Es una cadena que indica los atributos que queremos de la tabla. Ejem: 1\_3\_4\_2. Significa que obtendremos los atributos 1, 3,4 y 2 en ese orden, de la taba a la que se realiza la conexión.
4. Caract: Indica las características que tendrá el texto en el documento.
5. TiL: Tipo de Letra que se va utilizar.
6. TaL: Tamaño de letra.

❖ *Código a Leer:*

No	A	B	C	D ADO	E Sub	F Atrib	G Caract	H TiL	I TaL
1	0	4	Conexión1.Persona	1	3P2	1_	3J—F-N	Times..	12
2	0	4	Conexión2. Dirección	2		2_3_4	3J—F-N	Talon	12
3	0	4	Conexión1.Persona	1	3P1	5_2	3J—F-N	Arial	12

- A. Indica el nivel donde se encuentra el nodo.
- B. Indica el tipo de Nodo, que en este caso es 4.
- C. El texto que ve el usuario como descripción del nodo.
- D. EL ID de la conexión a la que nos referimos.
- E. La sub-tabla a la que nos referimos.
- F. Los atributos que queremos.
- G. Las características del texto en el documento.
- H. Tipo de letra.

## I. Tamaño de letra.

❖ *Procedimiento:*

1. Nos conectamos a la Conexión según el contenido de D.
2. Verificamos si E tiene datos.
  - Si tiene datos, buscamos los atributos en la posición determinada (xxPxx) de la tabla determinada (xxPxx), para después realizar la consulta apropiada.
  - Si no tiene datos, se traba con la tabla actual.
3. Tomamos los atributos contenidos en F en el orden en que se encuentran.
4. Verificamos sus características en G.
5. Tipo de letra de H.
6. Tamaño de letra de I.
7. Insertamos los datos en el Documento.

**Tipo “5” IF**

❖ *Objetivo:* Si la condición que se encuentra dentro del nodo, devuelve un valor verdadero, el intérprete, buscará interpretar los nodos siguientes, que tengan 1 nivel más alto y después los que estén en su mismo nivel, pero, si el valor del nodo es falso, el nodo entra directamente a los nodos que tengan su mismo nivel.

❖ *Atributos:*

1. Tipo1: Nos indica el tipo de dato que se encuentra del lado izquierdo de la condición. Existen 3 tipos diferentes para utilizar:
  1. Movimiento.
  2. DatoDeLaBase.
  3. Variable.
  4. Pregunta.
2. Tipo2: Nos indica el tipo de dato que se encuentra del lado derecho de la condición. Existen 4 tipos diferentes para utilizar:
  1. Movimiento.
  2. DatoDeLaBase.
  3. Variable.
  4. Texto o Numero
  5. Si o No.
3. Signo: los signos posibles para la comparación son: >, <, >=, <=, == y !=.

4. Mov1: Sí Tipo1 es 1, almacena un Nodo de **Tipo Movimiento**, para la comparación del dato Izquierdo.
5. Mov2: Sí Tipo2 es 1, almacena un Nodo de **Tipo Movimiento**, para la comparación del lado Derecho.
6. Dato1: Sí Tipo1 es 2, almacena un Nodo de **Tipo Dato de la Base**, para la comparación del lado Izquierdo.
7. Dato2: Sí Tipo2 es 2, hace lo mismo que el punto anterior, pero para el lado Derecho.
8. Variable1: Sí Tipo1 es 3, almacena un Nodo de **Tipo Variable**, para la comparación del lado Izquierdo.
9. Variable2: Sí Tipo2 es 3, almacena un Nodo de **Tipo Variable**, para la comparación del lado Derecho.
10. String1: Sí Tipo1 es 4, almacena una pregunta, a la cual se recibirá una respuesta por el usuario que sea forzosamente un Si o No.
11. String2: Sí Tipo2 es 4, almacena un texto o datos numéricos el cual será comparado con el lado izquierdo. Sí Tipo2 es 5, almacena un texto diciendo “True” o “False”, dependiendo de lo que seleccione el usuario.
12. Signo34: Sí Tipo2 es 1 o 3, se le asigna un signo a esta variable, el cual afectará el valor que devuelva **La Variable** o **El movimiento** con el atributo Número para después comparar el resultado con el lado izquierdo. Los signos son: +, -, \* y /.
13. Numero: Sí Tipo2 es 1 o 3, almacena un valor del 0 al 2,000,000.

❖ *Código a leer:*

NO	A	B	C	D	E	F	G	H	I	J
1	0	5	IF(..	1	1	Signo	Movimiento	Movimiento	Signo2	Número
2	0	5	IF(..	1	2	Signo	Movimiento	Variable	Signo2	Número
3	0	5	IF(..	2	2	Signo	DatoDeLaBase	DatoDeLaBase		
4	0	5	IF(..	2	3	Signo	DatoDeLaBase	Variable	Signo2	Número
5	0	5	IF(..	2	4	Signo	DatoDeLaBase	TextoNumero		
6	0	5	IF(..	3	1	Signo	Variable	Movimiento	Signo2	Número
7	0	5	IF(..	3	2	Signo	Variable	DatoDeLaBase		
8	0	5	IF(..	3	3	Signo	Variable	Variable	Signo2	Número
9	0	5	IF(..	3	4	Signo	Variable	TextoNumero		
10	0	5	IF(..	4	5	==	Pregunta	Si o No		

- I. Indica el nivel donde se encuentra el nodo.
- J. Indica el tipo de Nodo, que en este caso es 5.
- K. El texto que ve el usuario como descripción del nodo.
- L. El Tipo de Dato que se encuentra del lado izquierdo de la comparación. Dependiendo del valor de la casilla D, se cargan los atributos correspondientes a el Tipo de Dato en la casilla G.
- M. El Tipo de Dato que se encuentra del lado derecho de la comparación. Dependiendo del valor de la casilla E, se cargan los atributos correspondientes a el Tipo de Dato en la casilla H.
- N. Es el signo ( >, <, >=, <=, == ó != ) para realizar la comparación entre el resultado que refleje el contenido de G( lado izquierdo) y el contenido de H(

- lado derecho ). En el caso “10”, cuando la casilla G es Pregunta, el signo es forzosamente “==” pues el usuario sólo puede contestar Si o No.
- O. Es el conjunto de atributos que componen un TIPO DE DATO, que al interpretarlo, nos regresa el resultado del lado IZQUIERDO.
  - P. Es el conjunto de atributos que componen un TIPO DE DATO, que al interpretarlo, nos regresa el resultado del lado DERECHO.
  - Q. Sólo se carga cuando H es **Movimiento** ( E=1 ) o H es **Variable** ( E=3 ). Indica que operación matemática (+,-,\*ó/) se le aplica al **Movimiento** o **Variable** (Ejem. Var + 1 ó Var + 7 ), utilizando el contenido de la casilla J.
  - R. Es el número con el que se le aplica una operación matemática (+,-,\*ó/ ) a un **Movimiento** o **Variable**.

❖ *Procedimiento:*

1. *Verificar el Tipo1 de los atributos del Nodo.*
2. *Crear un nodo Temporal según el Dato obtenido en Tipo1, con las características tomadas de G.*
3. *Extraemos el Dato Resultante del Nodo Temporal.*
4. *Verificar el Tipo2 de los atributos del Nodo.*
5. *Crear un nodo Temporal según el Dato obtenido en Tipo1, con las Características tomadas de G.*
6. *Extraemos el Dato Resultante del Nodo Temporal.*
7. *Sí es Movimiento o Variable aplicamos la operación matemática según el contenido de I y J.*
8. *Comparamos según el signo.*
9. *Si la comparación es verdadera, verificamos si el NODO tiene nodos hijos o nodos de un nivel mayor, sí si los hay, entramos a los nodos hijos. Al terminar con los nodos hijos, entramos a el Nodo hermano de Nuestro NODO IF.*  
*Si la comparación es falsa, entramos al Nodo hermano de Nuestro NODO IF, sin tomar en cuenta sus nodos hijos.*

## Tipo “6” While

- ❖ *Objetivo:* Crear ciclos de repetición, dependiendo de la posición en donde se encuentre una consulta.
- ❖ *Atributos:*
  1. ADOI:El identificador de una de las conexiones que se encuentra en el Arreglo de Conexiones del Intérprete.
  2. POSI:la posición que estamos preguntando:

1. *Actual*
  2. *Inicio*
  3. *Anterior*
  4. *Siguiente.*
  5. *Ultima.*
  6. *Final*
3. TIPO: Son los Objetos con los que se puede comparar:
- i. *Conexión.*
  - ii. *Variable.*
  - iii. *Número.*
4. SIGNO: Son los signos de comparación:
1. >
  2. <
  3. >=
  4. <=
  5. !=
  6. ==
5. ADO2: Contiene el ID de conexión del Arreglo de conexiones. Únicamente se tiene datos TIPO es 1.
6. POS2: Tiene la posición del ADO2 a la que nos queremos referir. Sí TIPO es 2 tiene el identificador del arreglo de Variables. Sí TIPO es 3, contiene el Número con el que se indica la posición a comparar.

❖ *Código a Leer:*

No.	A	B	C	D ADO1	E Pos1	F Signo	G Tipo	H Pos2	I ADO2
1	0	6	Conex1.Act==Conex1.Final	1	1	6	1	6	1
2	0	6	Conex2.Act!=Var4	2	2	5	2	4	
3	0	6	Conex3.Act>5	3	4	1	3	5	

- A. Indica el nivel donde se encuentra el nodo.
- B. Indica el tipo de Nodo, que en este caso es 5.
- C. El texto que ve el usuario como descripción del nodo.
- D. El Identificado de Conexión.
- E. Posición
- F. Signo de comparación.
- G. El tipo de Objeto con el que se compara.



- H. La posición de ADO2 o El ID de Variable o Un numero Entero.
- I. El ID de la Conexión con la que se compara el ADO1.

❖ *Procedimiento:*

1. Tomamos la Conexión del Segmento D.
2. Nos movemos a la posición Indicada por E.
3. Vemos el Tipo de comparación en G.
  - a. Sí G es 1: Tomamos I como ID de Conexión y nos colocamos en la posición indicada en Pos2.
  - b. Sí G es 2: H el ID de Variable y tomamos el valor contenido en esa variable.
  - c. Sí G es 3: H se tomó como valor para comparar.
4. Comparamos los resultados según el signo colocado en F.
5. Si la comparación es True el intérprete entra a los Nodos hijos de While y después a los nodos del mismo nivel, pero si es False entra directamente a los nodos del mismo nivel sin tomar en cuenta a los nodos hijos del While.

## Tipo “7” Función

- ❖ *Objetivo:* Convertir un DatoDeLaBase, Variable o Fecha a Letra, Número, Moneda, etc...

❖ *Atributos:*

1. Función: Existen siete funciones diferentes:
  1. *Ordinal Masculino.*
  2. *Ordinal Femenino.*
  3. *Número a Letra.*
  4. *Número a Moneda.*
  5. *Fecha a Letra.*
  6. *Dato Tal Cual es.*
  7. *Código Postal.*
2. Tipo: El tipo de dato al que se le aplica la función:

1. *String*
2. *Variable.*
3. *DatoDeLaBase.*
3. String: Un carácter estático proporcionado por el usuario.
4. Var: El ID de la Variable.
5. Dato: Un Objeto de tipo DatoDeLaBase el cual convertiremos a la función determinada.
6. Caract: Características de cómo se mostrará el dato en el documento.
7. TiL: Tipo de Letra.
8. TaL: Tamaño de Letra.

❖ *Código a Leer:*

No.	A	B	C	D	E	F	G TiL	H TaL	I Caract
1	0	7	...	1	1	1234	Times..	12	3J—F-N
2	0	7	...	3	2	1	Times..	12	3J—F-N
3	0	7	...	4	3	DB	Times..	12	3J—F-N

- A. Indica el nivel donde se encuentra el nodo.
- B. Indica el tipo de Nodo, que en este caso es 7.
- C. El texto que ve el usuario como descripción del nodo.
- D. El tipo de Función.
- E. El tipo de Dato que entra en la función.
- F. El dato al que nos referiremos según el tipo.
- G. Tipo de Letra.
- H. Tamaño de Letra.
- I. Características.

❖ *Procedimiento:*

1. Verificamos el Tipo de Dato en E.
  - a. Tipo 1: Tomamos el dato Tal está en F.
  - b. Tipo 2: Tomamos el Dato del arreglo de Variables según el ID contenido en F.
  - c. Tipo 3: Toma un dato de la Base según los Datos contenidos en F.
2. El dato tomado lo enviamos a la función indicada en el campo D, y lo guardamos en una cadena de caracteres.
3. Indicamos el Tipo de Letra según G
4. El tamaño de letra según H.
5. Las Características según I.

6. Insertamos la **cadena de caracteres** al documento.

## Tipo “9” Comentario.

❖ *Objetivo:* Permitir al usuario tener un orden más completo sobre el contenido de su código.

❖ *Atributos:*

*Son los 3 atributos en común.(Nivel, Tipo y Texto).*

❖ *Código a Leer:*

No	A	B	C
1	0	9	Comentario

- A. Indica el nivel donde se encuentra el nodo.
- B. Indica el tipo de Nodo, que en este caso es 5.
- C. El texto que ve el usuario como descripción del nodo.

❖ *Procedimiento:*

1. *No se toma en cuenta para interpretar.*

## MODULO MANEJO DE TABLAS

Este módulo es de uso exclusivo para el administrador ya que el se encarga de definir que tablas (p. e. persona física, dirección) contendrá dicha operación (p. e. compra venta, testamento, etc.) y dicha tabla que rol(es) (p. e. vendedor, comprador, testador, etc.) jugará en dicha operación.

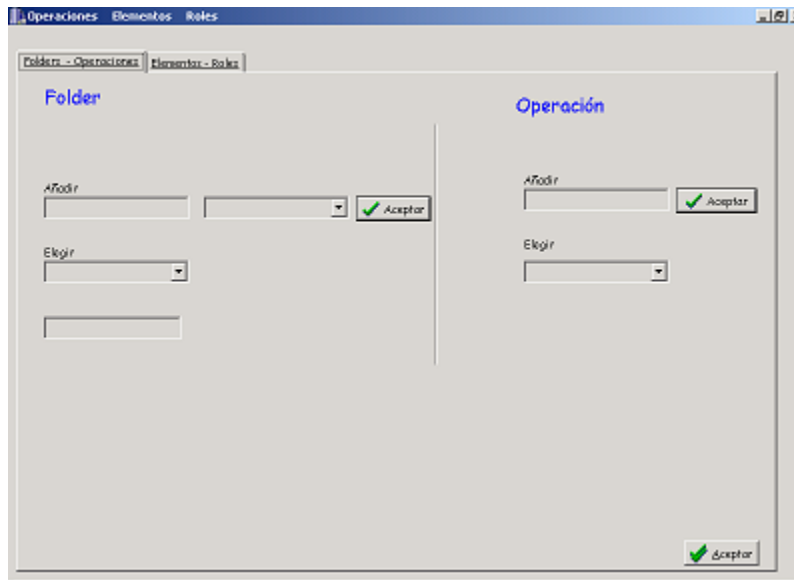


Figura 10. Interfaz Fólдер-Operación.

Como se puede ver en la Fig. anterior consta de dos partes el área del fólдер y la de operación, se puede tanto añadir así como, seleccionar tanto un fólдер como una operación.

### OPERACIÓN AÑADIR

Se escribe el nombre del fólдер o la operación a añadir en el componente edit con la etiqueta “Añadir”, en el caso del fólдер se selecciona de la lista desplegable que se encuentra a su lado derecho la tabla a asociar a dicho fólдер y en ambos casos se oprime el botón “Aceptar”.

### OPERACIÓN SELECCIONAR

Si sólo se necesita seleccionar ya sea un fólдер o una Operación se selecciona de la lista desplegable que encuentra debajo de la etiqueta “Elegir”.

Si ya se realizaron todas las operaciones necesarias en esta pantalla se oprime el botón que se encuentra en la esquina inferior derecha con la leyenda “Aceptar” este botón nos manda a la siguiente pantalla:

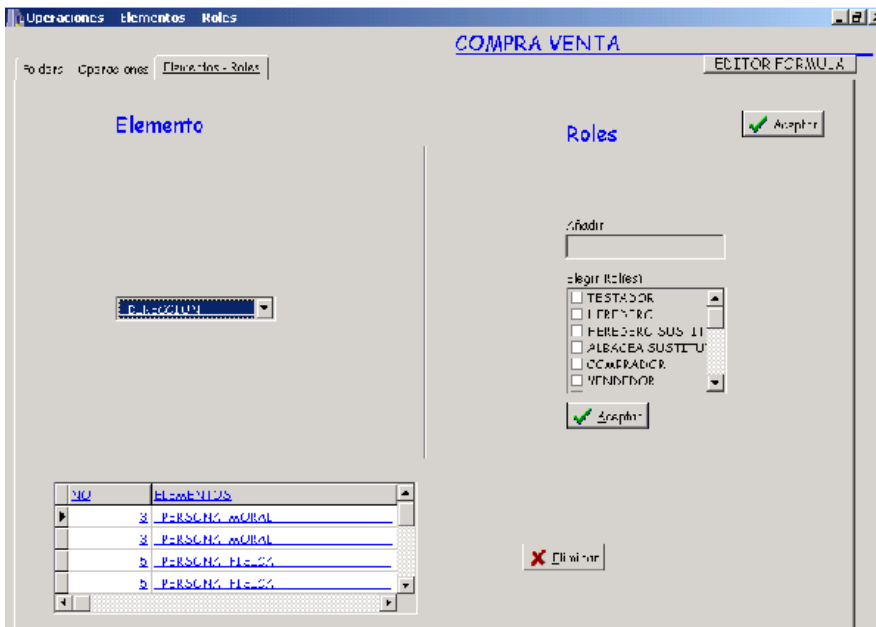


Figura 11. Interfaz Elementos-Roles.

Como en la pantalla anterior ya seleccionamos la Operación aparece el nombre en la parte superior y en la rejilla de la parte inferior aparecen las tablas asociadas ya con anterioridad con sus respectivos roles asociados, en el caso de no tener tablas asociadas con sus roles aparece vacía.

En la Fig., anterior se pueden observar dos secciones una de “Elementos” y otras de “Roles”, podemos añadir Elementos a la operación y añadir Roles a la base de datos en la Tabla Roles y a su vez asociar o eliminar Roles a los Elementos.

## OPERACIÓN AÑADIR

Cuando deseamos añadir una tabla o un rol a dicha tabla, de una Operación dada, se selecciona en la lista desplegable que se encuentra en la sección de “Elementos” bajo la etiqueta de “Elegir”.

Cuando deseamos añadir un rol a la tabla de roles de la base de datos insertamos el nombre que le queremos designar en la sección de “Roles” en el campo de edición que se encuentra debajo de la etiqueta “Añadir” y oprimimos el botón que se encuentra a su derecha con la leyenda “Aceptar”.

## OPERACIÓN ASOCIAR

Cuando deseamos asociar rol(es) a un Elemento, se selecciona dicho elemento de la lista desplegable de la sección de “Elementos” y se seleccionan a su vez los roles que se quieren asociar de la lista que se encuentra bajo la etiqueta roles y se oprime el botón “Aceptar” que se encuentra debajo.

## OPERACIÓN ELIMINAR

Cuando deseamos eliminar tablas con sus respectivos roles se selecciona la tabla a eliminar de la rejilla y se oprime el botón con la leyenda “Eliminar”.

## CREACIÓN DE TABLAS

Para crear las tablas que utilizará el usuario, el administrador debe entrar a dicho módulo y en el campo “Nombre tabla” insertar el nombre de la nueva tabla.

Todas las tablas que se generen en este módulo son creadas por el administrador del sistema, todo de acuerdo a los requerimientos del usuario.

Nombre Campo	Tipo	Tamaño
Nombre	char	20

Con "Reanudar", se refrescan los componentes, para aceptar nuevos datos

Figura 11. Interfaz Crear tabla simple

Al presionar el botón de “Crear tabla” se crea la tabla físicamente en la base de datos además que se da de alta en la tabla “ELEMENTOS”, la cual corresponde a las tablas de control del sistema y se le asigna un “ID\_TABLA”.

Al presionar el botón “Agregar campo”, se insertan todos los campos en la tabla ya creada, esto utilizando la sentencia SQL, **alter table**, lo cual nos permite insertar campos nuevos en una base de datos ya creada.

Esto se hizo porque al no saber que campos integrará el usuario en la tabla, se crea una tabla con el campo “ID\_MIEMBRO”, y se le da al usuario la libertad de insertar el número y tipo de datos que requiera.

Todas las tablas al crearse en la base de datos a su nombre se le agrega un guión bajo “\_”, el cual permite identificar cuales son las tablas generadas por el usuario, y así, evitar confusiones con las tablas de control del sistema.

El “ID\_MIEMBRO” de las tablas es de tipo float, y se genera automáticamente por el sistema, cada vez que se agrega un dato nuevo a los campos de la tabla, así, evitamos que el administrador tenga problemas con ellos.

Ya creadas las tablas, sólo el administrador podrá ver todos los registros que hay en ella, esto para que ningún usuario pueda modificar datos por error, como el “ID\_MIEMBRO”, que es el que utiliza el sistema para establecer las relaciones con las tablas creadas por el administrador.

## TABLAS COMPUESTAS

Existe un tipo especial de tablas, las tablas compuestas, las cuales nos permiten relacionar los datos que existen en las tablas generadas por el usuario, esto, sin tener que crear relaciones previas entre ellas.

Una tabla compuesta esta formada por los ID\_MIEMBRO de los datos a los que se hace referencia.

En este caso, “\_PERSONAFISICA” es una tabla compuesta, donde tenemos como nombres de los campos el “ID\_TABLA”, de cada tabla, junto con el nombre de la tabla, así, sabemos a que tabla se hace referencia.

Para crear esta tabla, se relacionaron las tablas “\_NOMBRES” y “\_DIRECCION”, de donde tomamos sólo los ID\_MIEMBRO de cada fila, así, mediante la tabla compuesta, podemos hacer referencia a todos los datos que se requieren.

4:Data in Table '\_NOMBRES'

ID_MIEMBRO	NOMBRE	APELLIDOP	APELLIDO
1	Carol	Soto	Gomez
2	Yessika	Sanchez	Borjas
3	Roberto	de la Barrera	Pliego
*			

3:Data in Table '\_DIRECCION'

ID_MIEMBRO	Estado	Municipio	Colonia	Numero
12	Mexico	Ecatepec	Rio de Luz	S/N
13	Mexico	Tultitlan	Fuentes del Valle	178
*				

*Figura 12. Tablas simples*

Así, en este caso, tenemos que, según la tabla compuesta...

2:Data in Table '\_PERSONAFISICA'

ID_MIEMBRO	NOMBRES	DIRECCION
1	1	12
2	2	13
*		

*Figura 13.Tabla compuesta*

La persona 2, Carol Soto Gómez, tiene la dirección indicada por el ID\_MIEMBRO =12.

Para crear la tabla compuesta, se asigna un nombre a la tabla, donde de la misma forma que en el caso anterior se crea la tabla con el campo “ID\_MIEMBRO”, se da de alta en la tabla “ELEMENTOS”, esto al presionar el botón “Crear tabla”.



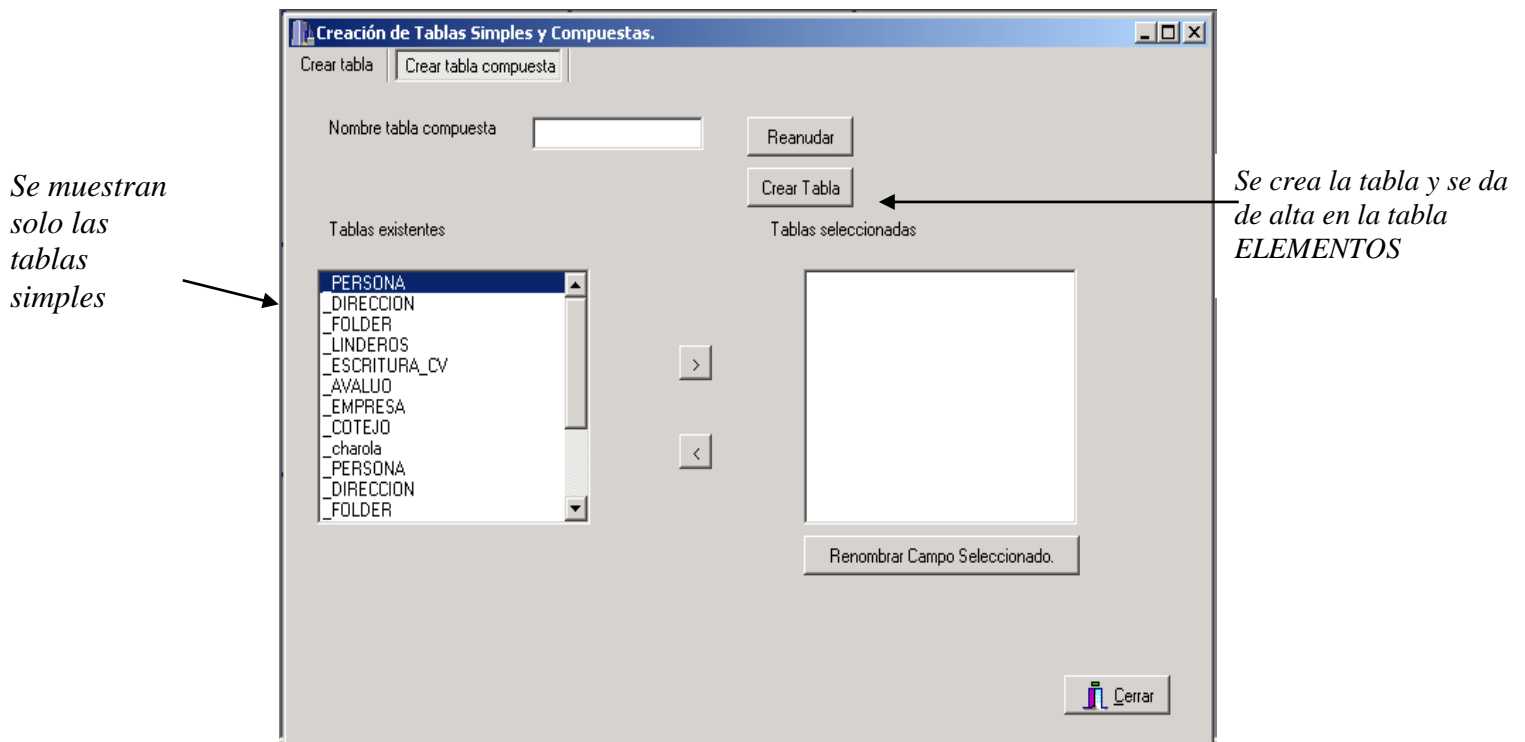


Figura 14. Interfaz crear tabla compuesta

Se seleccionan las tablas que se quieren relacionar, estas van a ser las que den el nombre a los campos que formen la base.

Al presionar el botón “Relacionar tablas”, se agregan los campos a la tabla, además de que a las tablas en el campo “T1”, de la tabla “ELEMENTOS” se agrega un “1”, el cual nos va a indicar que tenemos una tabla compuesta y que tiene que hacer referencia a otras tablas.

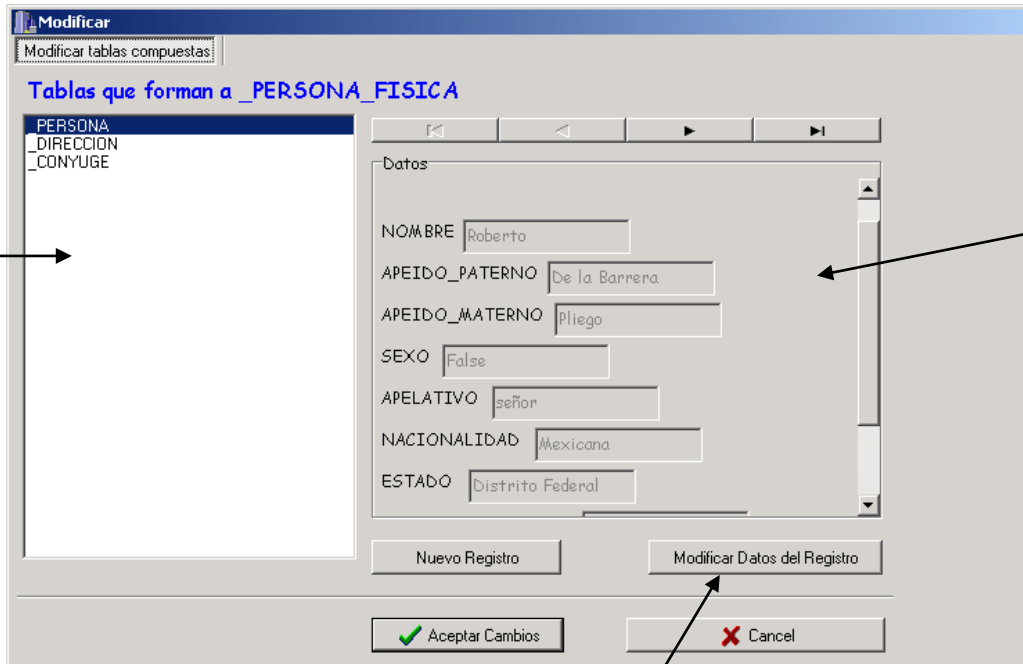
Al capturar los datos, se insertarán en la tabla correspondiente, al mismo tiempo, los **ID\_MIEMBRO** de cada campo, si la tabla es compuesta, se agregarán en la tabla compuesta de la que forman parte, así, no existe problema en que se relacionen los datos.

## MODIFICAR TABLAS COMPUESTAS



Figura 15. Botón Modificar

Se creó este módulo para la modificar datos que el usuario ya tenga en cierto fólder, aquí podemos seleccionar un contenido diferente de los campos que tenemos en la lista de las tablas que forman a la tabla compuesta.



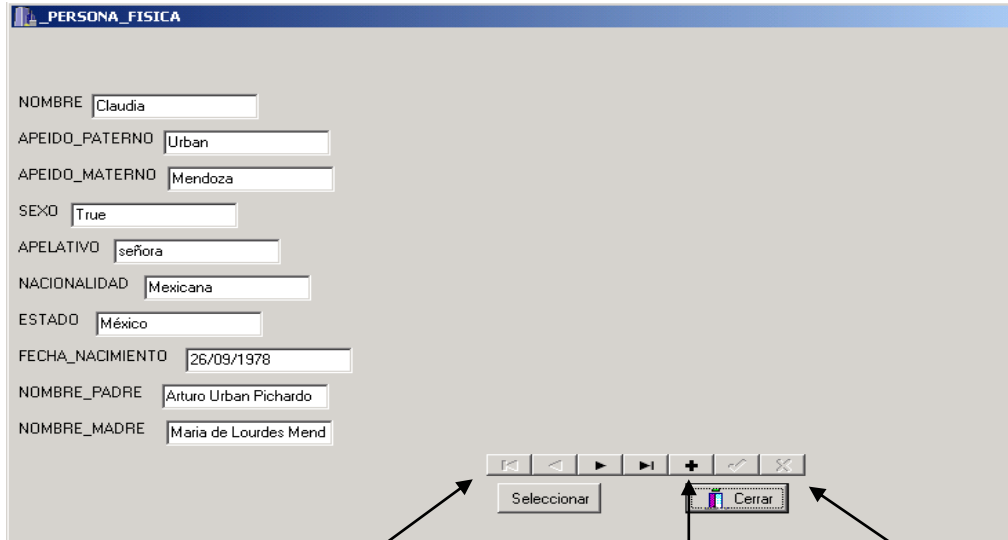
Tablas que forman a la tabla compuesta

Aquí se muestran los datos actuales

*Figura 16. Interfaz Modificar tablas*

Al presionar el botón “Modificar Datos del Registro”, se muestra una forma que se genera dinámicamente y que muestra todos los campos de la tabla seleccionada, con la posibilidad de poder recorrer los registros que ésta contiene además de poder insertar nuevos registros.

Campos de la tabla



Recorrer registros de la tabla

Insertar registro

Cancelar inserción de nuevo registro

*Figura 17. Modificar datos del registro*

Al presionar el botón “Seleccionar”, los datos que se estén mostrando en ese momento, se muestran ahora como datos actuales.

The screenshot shows a software window titled "Modificar" with a sub-tab "Modificar tablas compuestas". The main heading is "Tablas que forman a \_PERSONA\_FISICA". On the left, a list of tables includes "PERSONA", "\_DIRECCION", and "\_CONYUGE", with "PERSONA" selected. The right side contains a "Datos" section with several text input fields: "NOMBRE" (Claudia), "APELLIDO\_PATERNO" (Urban), "APELLIDO\_MATERNO" (Mendoza), "SEXO" (True), "APELLATIVO" (señora), "NACIONALIDAD" (Mexicana), and "ESTADO" (México). Below the fields are two buttons: "Nuevo Registro" and "Modificar Datos del Registro". At the bottom, there are two buttons: "Aceptar Cambios" (with a green checkmark icon) and "Cancel" (with a red X icon).

Figura 18. Modificar tabla compuesta con datos ya modificados.

## CAPITULO 4 MARCO TEORICO

### BASES DE DATOS

#### Evolución histórica

Actualmente, los SGBDs constituyen el núcleo fundamental del soporte lógico a los sistemas de información. Desde la aparición de los primeros SGBDs comerciales en la década de los 60 hasta la actualidad, se han sucedido tres generaciones distintas de SGBDs basadas en tres modelos de datos diferentes, siempre basadas en modelos de datos o modelos de procesos del Sistema de Información.

Los tres modelos en los que se ha basado el desarrollo de las bases de datos son el jerárquico, en red y relacional.

El modelo jerárquico dominó el mercado de los SGBD hasta mediados de los 80.

Durante este mismo período, surgió el modelo en red con el que se pretendía sustituir a los SGBDs jerárquicos, lo que no se consiguió. A principios de los ochenta el modelo jerárquico comenzó a ser sustituido por una nueva generación de SGBD basados en el modelo relacional que, en la actualidad, dominan ampliamente el mercado. De esta manera, los modelos jerárquico y de red han caído en desuso, si bien siguen siendo empleados en aquellos sistemas donde se ha conseguido obtener un elevado rendimiento y no nacen nuevas necesidades en el Sistema de Información.

Los SGBDs jerárquicos fueron los primeros en aparecer. Una base de datos jerárquica se puede visualizar como una estructura en árbol. Las bases de datos jerárquicas son bastante rígidas. Una vez diseñada la base de datos, es complejo cambiarla y, además, es necesario un conocimiento amplio de la forma en la que se han almacenado los datos para poder recuperarlos de forma efectiva. Por ello, a pesar de dominar el mercado de SGBDs en sus comienzos, han ido decayendo y actualmente no se encuentran en el mercado.

Los SGBDs en red fueron una evolución del modelo jerárquico. En una base de datos en red, cada uno de los registros están enlazados entre sí, pero, no necesariamente siguiendo

una estructura en árbol. El modelo en red elimina parte de las rigideces del modelo jerárquico pero aumenta la complejidad para modificar la estructura de la base de datos. Por ello, a pesar de su buen rendimiento, el número de instalaciones con SGBDs en red siempre ha sido pequeño y, hoy en día, tampoco se encuentran en el mercado. Sin embargo, aún quedan instalaciones basadas en estos dos modelos de datos que responden con gran eficiencia y plena satisfacción de sus usuarios.

Los SGBDs relacionales. En una base de datos relacional se representan los datos como un conjunto de tablas bidimensionales compuestas de filas y columnas. Cada fila representa una relación entre un conjunto de valores y está identificada por una clave única. Los SGBDs relacionales son muy flexibles y de fácil manejo, lo que los ha convertido en el modelo dominante en la actualidad. Un factor decisivo en la implantación de los SGBDs relacionales ha sido el lenguaje SQL (Structured Query Language) para la interrogación y el manejo de datos del modelo relacional, unido con la normalización propuesta por Codd para estos tipos de SGBDs.

Los Sistemas de Información que, debido a sus características (inclusión de componentes multimedia, conectividad a través de Internet, etc.) se modelizan siguiendo un esquema conceptual orientado a objetos son susceptibles de emplear un SGBD objetual, que se adapta para emplear una implementación relacional, subyaciendo por tanto un SGBD relacional por debajo. De esta manera, se puede separar entre los SGBD puramente objetuales de aquellos SGBD relaciones con soporte para la orientación a objetos.

Es decir, la diferencia de una BD respecto a otro sistema de almacenamiento de datos es que éstos se almacenan de forma que cumplan tres requisitos básicos:

No redundancia: Los datos se almacenan una sola vez. Si varias aplicaciones necesitan los mismos datos no crearán cada una su propia copia sino que todas accederán a la misma.

Independencia: Los datos se almacenan teniendo en cuenta la estructura inherente a los propios datos y no la de la aplicación que los crea. Esta forma de trabajar es la que permite que varias aplicaciones puedan utilizar los mismos datos. Se puede hablar de dos tipos de independencia: independencia física, de tal manera que la estructura física de la BD puede ser modificada de forma transparente para los programas que la utilizan, e independencia lógica, es decir el programador usa la BD pero desconoce su estructura interna.

Concurrencia: Varios usuarios, ejecutando la misma o diferente aplicación, podrán acceder simultáneamente a los datos.

### Funciones de un SGBD

Un SGBD debe proporcionar un amplio surtido de funcionalidades para poder cumplir Adecuadamente su misión. Normalmente se clasifican en definición, manipulación y utilización.

**Función de definición:** Permite describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del SGBD denominada lenguaje de definición de datos (LDD o DDL, Data Definition Language).

**Función de manipulación:** Permite buscar, añadir, suprimir y modificar los datos de la BD.

Esta función es realizada por una parte del SGBD denominada lenguaje de manipulación de datos (LMD o DML, Data Manipulation Language).

**Función de utilización:** Incluye otras funcionalidades tales como: modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, rearranque, protección frente a accesos no autorizados, gestión de la concurrencia, estadísticas de utilización, etc.

### **Soporte ODBC**

ODBC (siglas que significan Open DataBase Connectivity, Conectividad Abierta de Bases de Datos) se define como un método común de acceso a bases de datos, diseñado por Microsoft para simplificar la comunicación en Bases de Datos Cliente/Servidor. ODBC consiste en un conjunto de llamadas de bajo nivel que permite a las aplicaciones en el cliente intercambiar instrucciones con las aplicaciones del servidor y compartir datos, sin necesidad de conocer nada unas respecto a las otras.

Las aplicaciones emplean módulos, llamados controladores de bases de datos, que unen la aplicación con el SGBD concreto elegido. Se emplea el SQL como lenguaje de acceso a los datos. El SGBD debe proporcionar los controladores adecuados para poder ser empleados por los distintos lenguajes de programación que soporten ODBC.

## **VENTAJAS EN EL USO DE BASES DE DATOS.**

La utilización de bases de datos como plataforma para el desarrollo de Sistemas de Aplicación en las Organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentarán a continuación:

- \* Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- \* Eliminación de información inconsistente: si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas.
- \* Permite compartir información
- \* Permite mantener la integridad en la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos.

## **ADO**

ADO es un objeto programable que puede acceder datos en una base de datos.

En general existen los siguientes objetos ADO's cada uno con sus propiedades y métodos:

**COMMAND.-** Se usa para ejecutar un query (instrucción especializada de SQL) contra la base de datos, este query puede ser para crear, agregar, recuperar, borrar o actualizar registros o renglones.

**CONNECTION.-** Se usa para crear una conexión o enlace o canal a la base de datos, usando esta conexión o enlace se puede procesar la base de datos.

**ERROR.-** Este objeto convendrá los posibles errores que se generen al usar la base de datos.

FIELD.- Este objeto permite manipular o procesar una columna de la tabla de la base de datos, es necesario aclarar que este objeto no trabaja directamente con la tabla en disco sino trabaja contra una copia de la misma llamada RECORDSET o tabla virtual en memoria.

PARAMETER.- Almacena información sobre algún parámetro que se usa en PROCEDIMIENTOS ALMACENADOS o QUERYS, un procedimiento almacenado (stored procedure) es un conjunto de instrucciones almacenadas que procesan o manipulan la base de datos.

RECORD.- Se utiliza para almacenar y procesar un renglón de información de un RECORDSET.

RECORDSET.- Es el objeto ADO mas importante, almacena y procesa una tabla completa de una base de datos en memoria incluyendo información extra de la tabla.

Recordar que una tabla en disco o su representación en memoria (EL RECORDSET) es un conjunto de renglones (records) y columnas (FIELDS).

STREAM.- Es un objeto especializado en escribir flujos de datos de tipo texto o binarios a archivos de datos en discos (y para el próximo curso ya estará incluido este tipo de archivos en su propia unidad en este curso).

Los componentes utilizados por nosotros usan ADO, que son unos componentes ActiveX mediante los cuales se representan elementos de alto nivel: conexiones con las bases de datos, comandos a ejecutar, conjuntos de datos.

ADO es una capa que podríamos llamar de alto nivel, al abstraer elementos muy concretos de la base de datos, lo cual facilita el trabajo del desarrollador.<sup>§</sup>

El acceso a los datos, no se realiza físicamente a través de los componentes ADO, estos, a su vez, se delegan en otros, conocidos como proveedores OLE DB.

En el ámbito OLE DB pueden distinguirse 2 objetos: el proveedor de datos y el consumidor de datos. Adicionalmente, OLE DB provee otros servicios para la comunicación entre esto 2 objetos.

Al utilizar ADO, aislamos nuestro programa de la necesidad de usar directamente OLE DB. En nuestro caso, además, usando los componentes ADO de C++Builder tampoco tenemos que comunicarnos con ADO.

---

<sup>§</sup> Charle Ojeda. "Programación con C++Builder".



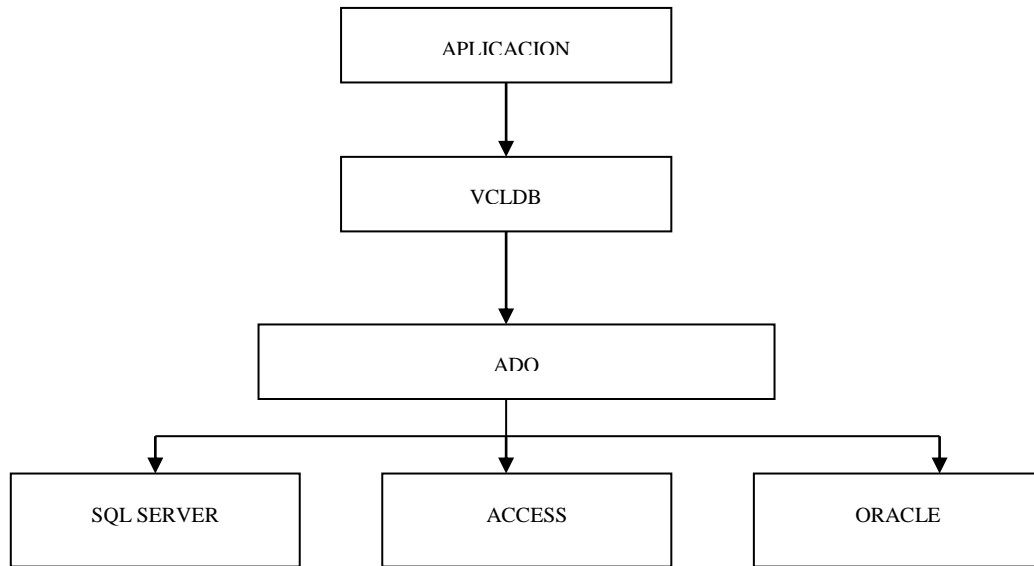


Figura .Estructura de bloques ADO/OLE DB §

## SQL

La historia de SQL (que se pronuncia deletreando en inglés las letras que lo componen, es decir “ese-cu-ele” y no “Siquel” como se oye a menudo) empieza en 1974 con la definición, por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación de IBM, de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional. Este lenguaje se llamaba SEQUEL (Structured English Query Language) y se implementó en un prototipo llamado SEQUEL-XRM entre 1974 y 1975. Las experimentaciones con ese prototipo condujeron, entre 1976 y 1977, a una revisión del lenguaje (SEQUEL/2), que a partir de ese momento cambió de nombre por motivos legales, convirtiéndose en SQL.

El prototipo (System R), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos. Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías empezaron a desarrollar sus productos relacionales basados en SQL.

A partir de 1981, IBM comenzó a entregar sus productos relacionales y en 1983 empezó a vender DB2. En el curso de los años ochenta, numerosas compañías (por ejemplo Oracle y Sybase, sólo por citar algunos) comercializaron productos basados en SQL, que se convierte en el estándar industrial de hecho por lo que respecta a las bases de datos relacionales.

En 1986, el ANSI adoptó SQL (sustancialmente adoptó el dialecto SQL de IBM) como estándar para los lenguajes relacionales y en 1987 se transformó en estándar ISO. Esta

§ Charte Ojeda. “Programación con C++Builder”.Pag.605.

versión del estándar va con el nombre de SQL/86. En los años siguientes, éste ha sufrido diversas revisiones que han conducido primero a la versión SQL/89 y, posteriormente, a la actual SQL/92.

El hecho de tener un estándar definido por un lenguaje para bases de datos relacionales abre potencialmente el camino a la ínter comunicabilidad entre todos los productos que se basan en él. Desde el punto de vista práctico, por desgracia las cosas fueron de otro modo. Efectivamente, en general cada productor adopta e implementa en la propia base de datos sólo el corazón del lenguaje SQL (el así llamado Entry level o al máximo el Intermediate level), extendiéndolo de manera individual según la propia visión que cada cual tenga del mundo de las bases de datos.

Actualmente, está en marcha un proceso de revisión del lenguaje por parte de los comités ANSI e ISO, que debería terminar en la definición de lo que en este momento se conoce como SQL3. Las características principales de esta nueva encarnación de SQL deberían ser su transformación en un lenguaje stand-alone (mientras ahora se usa como lenguaje hospedado en otros lenguajes) y la introducción de nuevos tipos de datos más complejos que permitan, por ejemplo, el tratamiento de datos multimediales.<sup>§</sup>

La gramática sintáctica utilizada en las diferentes lecciones para explicar la sintaxis de las instrucciones SQL es muy simple:

- El texto en mayúscula tiene que aparecer como es. El lenguaje SQL no es case-sensitive, por lo que después, en los ejemplos, aparecerán tanto letras mayúsculas como minúsculas, de manera que aumenta la legibilidad.
- El texto en minúscula indica elementos que tienen que ser especificados después. Por ejemplo, si aparece `vínculos_de_columna` no significa que en esa posición haya que escribir correctamente la secuencia de caracteres, sino que encontrarán sitio los vínculos de la columna, especificados con su sintaxis concreta, que se explica en otro lugar.
- Los corchetes (`[]`) indican elementos opcionales y que por tanto no tienen por qué aparecer.
- Los puntos suspensivos (`...`) indican elementos que se pueden repetir. Por ejemplo:

`[, [vínculo_de_tabla] ...]`

indica que el elemento “`[vínculo_de_tabla]`” se puede repetir tantas veces como sea necesario.

- Elementos separados por el carácter “`|`”, y eventualmente agrupados entre llaves (`{ }`), indican elementos que alternativos. Por ejemplo:

<sup>§</sup> <http://www.asptutor.com/sql/sql.asp/>

*{ elemento1 | elemento2 }*

indica que en esa posición habrá que escribir o elemento1 o elemento2.

- El texto escrito entre comillas simples (‘) o dobles (“) se escribe exactamente como está indicado (comillas, mayúsculas y minúsculas incluidas).
- Los demás caracteres (por ejemplo las comas (,) o las comillas dobles (“)) tienen que aparecer como son.

#### Guía de características de SQL Server 7.0

SQL Server 7.0 hace pasos agigantados en rendimiento, **confiabilidad y escalabilidad**, dando a su organización muchas oportunidades para crear soluciones inteligentes de negocios del mundo real. Al poner de manifiesto una necesidad de características más simplificadas y que constituyan un ahorro, las organizaciones han inspirado las siguientes innovaciones en SQL Server 7.0:

- Escalable desde una computadora portátil hasta un cluster de multiprocesadores
- Bloqueo dinámico a nivel de fila
- Administración automática dinámica
- Amplia variedad de opciones de replicación CONFIDENCIAL
- SQL Server Desktop
- Servicio OLAP integrados
- Servicios de transformación de datos
- Microsoft English Query
- Microsoft Repository
- Integración con Microsoft Office 2000

SQL Server 7.0 está diseñado para acomodar con facilidad más datos, transacciones y usuarios.

Escalable desde una computadora portátil hasta un cluster de multiprocesadores. Entre las innovaciones en esta edición está la capacidad de SQL Server para escalar a las necesidades de su organización. Con SQL Server 7.0, las aplicaciones en las que su organización se apoya se pueden adaptar y acomodar hasta terabytes de datos y miles de usuarios.

Procesador de consultas. SQL Server 7.0 también incluye un procesador de consultas rediseñado que proporciona poderoso soporte para las grandes bases de datos y consultas complejas.

Bloqueo dinámico a nivel de registro. SQL Server 7.0 ahora da soporte al bloqueo dinámico de filas, haciéndolo la mejor elección para sistemas de OLAP y de datawarehousing. El administrador de bloqueo ajusta dinámicamente los recursos que usa para grandes bases de datos. Esto significa que ya no necesita ajustar manualmente la configuración de bloqueo del servidor. §

---

§ <http://www.sql-server-performance.com/>

**Bibliografía Básica:****Ingeniería de Software**

Roger S. Pressman  
Editorial McGraw Hill

**Ingeniería de Software**

Kendall y Kendall

**Programación con C++Builder 5**

Francisco Charte Ojeda  
Editorial Anaya.

**Bibliografía Complementaria****Lado Oscuro de C++ Builder**

Dr. Ian Marteens  
<http://www.marteens.com>

**Análisis estructurado de sistemas**

Chris Gane & Trish Sarson  
Librería “El Ateneo” Editorial

<http://www.marteens.com>

<http://www.thebits.org>

<http://www.bytaminc.com>

<http://www.ciudadfutura.com/cworld/>

<http://www.asptutor.com/sql/sql.asp/>

<http://www.sql-server-performance.com/>

<http://www.temporaldoorway.com/programming/cbuilder/>