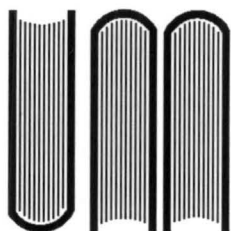


D
DICIEMBRE
\$250 =

UNIVERSIDAD DE MONTERREY

DIVISIÓN DE INGENIERÍA, CIENCIAS NATURALES Y EXACTAS



UNIVERSIDAD
DE MONTERREY

HCA para la Verificación de Pago de Nómina en CEMTEC S.A. de C.V.

PROGRAMA DE EVALUACIÓN FINAL QUE PRESENTA:

040.0016
R 696h
1997

Héctor Javier Rodríguez Mijares

en opción al Título de:

902873

INGENIERO EN SISTEMAS COMPUTACIONALES

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

Dedicatorias

A mis padres,

Héctor J. Rodríguez González y María del Consuelo Mijares de Rdz.

En compensación a todas esas noches de desvelo y cansancio estando a mi lado en los momentos más difíciles de mi carrera; aquí está el resultado de una inversión que NADIE me podrá quitar.

A Adriana,

como promesa de hacer todo lo humanamente posible por hacerla feliz, en respuesta a su cariño y comprensión en los momentos en que estuve más agobiado; gracias por no haber perdido la fe en mí ni un instante, gracias por ser una novia ejemplar.

A todo aquel que tiene sueños y que lucha tenazmente por alcanzarlos, que deposita su confianza en Dios y se aferra a Él aún y cuando todo parezca estar en contra. Esta es la prueba de que SÍ se puede,

No te desanimes, mira al frente y camina de la mano del Señor.

Agradecimientos

A Dios, mi Señor, porque una vez más me has demostrado que eres fiel y que cumples Tus promesas.

Porque me has dado una familia maravillosa y me has llenado de bendiciones. *Mi copa está rebozando...*

A mi abuelo, Don Gilberto Rodríguez Zamarripa,

Gracias a ti soy lo que he llegado a ser y seré mucho más,
que a los 83 años tenga yo el vigor y ánimo que posees
¡Que Dios te bendiga abuelo, y te conserve muchos años más!

A mi asesor, Antonio Balboa (Toño).

Gracias por tu tiempo, tus conocimientos y tu amistad.
Tus atinados consejos enriquecieron mi labor y se verán reflejados
en beneficios para todo aquel que tenga contacto con ésta.

PRÓLOGO

Actualmente el crecimiento de la tecnología de grupo, de trabajo en equipo, de las comunidades virtuales y la era del documento electrónico, han dado paso al manejo de bases de documentos como complemento de las áreas no cubiertas por las bases de datos tradicionales, proporcionándole a los Negocios una amplia gama de posibilidades de eficientizar sus procesos de administración y control, así como de estar gradualmente cambiando la cultura de información basada en el manejo de papel, hacia la cultura de cero papeles.

En este sentido una de las herramientas de mayor potencial en el manejo de documentos electrónicos y excelente aliada para el fomento de la cultura de cero papeles, es "Lotus Notes". No obstante, para las Negocios con amplia dispersión geográfica presenta dificultades de infraestructura en su alcance; en tales circunstancias, las herramientas utilizadas en la red global pública de comunicaciones, el Internet y las utilizadas recientemente de manera particular con coberturas limitadas sin accesar la red global pública, el Intranet, resultan ser excelentes opciones mediante el uso los Browsers, para los Negocios con las características de dispersión geográfica. Por tanto, las dos posibilidades existen, lo complicado es comunicar ambas y construir aplicaciones que permitan explotarlas y proporcionen sus respectivas bondades.

El presente documento, contiene el desarrollo de una solución en la cual Héctor J. Rodríguez M. emplea la tecnología de manejo de bases de documentos de "Lotus Notes" y de Internet e Intranet de "MicroSoft Explorer", combinándolas con la tecnología de "Domino" para satisfacer la necesidad del manejo electrónico de la Verificación del Pago de Nómina, mediante una Herramienta de Coordinación de Acciones desarrollada para Cemtec S.A., con el objeto de apoyar el proceso de entrega y verificación del recibo de nómina en los Negocios. Con ello, se dará la flexibilidad a todos los empleados de consultar, aprobar mediante una firma electrónica y/o solicitar aclaraciones sobre sus pagos de nómina, desde la comodidad de su oficina utilizando su cuenta de "Lotus Notes", o bien desde cualquier otro negocio filial de Cemex utilizando Intranet y desde cualquier otra parte del mundo utilizando Internet

Héctor J. Rodríguez M. expone y detalla a través del documento, los distintos elementos que se han integrado en la solución, destacando los conceptos de Cliente Servidor (C/S), Conectividad de Bases de Datos Abiertas (ODBC), Manejo de Documentos Electrónicos, Trabajo en Grupo (Groupware), Internet, Intranet y Firma Electrónica. De igual manera, explica los principales conceptos de la metodología de análisis, diseño y desarrollo orientado a Objetos que ha utilizado, incluyendo los principales entregables de cada una de sus etapas en distintos capítulos del documento.

Gracias al excelente manejo de los conceptos de los elementos antes mencionados y a la aplicación de la metodología de Objetos bajo un matiz de su estilo muy personal, así como del uso de técnicas de Prototipeo durante la construcción de la solución, Héctor J. Rodríguez M. ha producido una excelente Aplicación de apoyo a la administración y control del proceso de verificación del pago de nómina en el Negocio y ha convertido en todo un éxito el desarrollo del presente proyecto.

Ing. Antonio Balboa Garza
Noviembre de 1997.

ÍNDICE

INTRODUCCIÓN	1
1. CEMTEC	4
1.1 Antecedentes	5
1.2 Misión de CEMTEC	5
1.3 CEMTEC y el Servicio al Cliente	5
1.4 Logros de CEMTEC	6
1.5 Productos y Servicios	7
1.5.1 Sistemas de Información Ejecutiva	7
1.5.2 Herramienta para la Coordinación de Acciones (Workflow)	8
1.5.3 Automatización de Procesos	8
1.5.4 JDEdwards	8
1.5.5 Groupware	8
1.5.6 Internet	9
1.5.7 Networking	9
1.5.8 Network & System Management	9
1.5.9 Telecomunicaciones	9
1.5.10 Telefonía	9
1.5.11 Herramientas de apoyo Logístico	9
1.5.12 Multimedia	10
1.5.13 Centro de Educación	10
1.5.14 Help Desk	10
2. HERRAMIENTA PARA LA COORDINACION DE ACCIONES (HCA)	11
3. ESTUDIO PRELIMINAR	14
3.1 Antecedentes y descripción del problema	14
3.1.1 Sobre el proceso actual	16
3.1.2 Identificación del Problema	18
3.2 Solución propuesta al problema	19
3.2.1 Características del proceso	20
3.2.2 Requerimientos de la herramienta	22
3.2.3 Características de la herramienta	23
3.2.4 Ventajas que se pretende obtener	24
3.2.5 Dificultades que se pueden presentar	25

4. DELIMITACIÓN DEL PROYECTO	26
4.1 Objetivo General del Proyecto	26
4.2 Alcance	27
4.3 Objetivos Particulares	27
4.3.1 Objetivos personales	27
4.3.2 Objetivos del Sistema	28
4.4 Recursos necesarios	29
4.4.1 Hardware	29
4.4.2 Software	30
4.4.3 Literatura	30
4.4.4 Tiempo	30
5. MARCO TEÓRICO	31
5.1 TECNOLOGÍA CLIENTE / SERVIDOR	31
5.1.1 Conceptos	33
5.1.2 De la teoría, a la práctica	35
5.1.3 Características del ambiente cliente / servidor	36
5.2 Consideraciones para el desarrollo [RENAUD, 93]	39
5.3 MODELO ORIENTADO A OBJETOS	40
5.3.1 Estructura de Objetos.	40
5.3.2 Jerarquía de Clases.	42
5.3.3 Herencia.	43
5.3.4 Beneficios de la tecnología orientada a objetos: [MARTIN, 93]	44
5.4 ANALISIS Y DISEÑO ORIENTADO A OBJETOS	46
5.4.1 Dos Tipos De Modelos [Martin, 93]	47
5.4.2 Análisis de la Estructura de Objetos [Martin, 93]	48
5.4.3 Análisis del Comportamiento de objetos	49
5.4.4 Diseño de la estructura y el comportamiento de un objeto (DEO y DCO)	51
5.5 LOTUS NOTES	52
5.5.1 Aplicaciones de Lotus Notes	52
5.5.2 Tipos de aplicaciones de Notes	53
5.5.3 Aplicaciones de <i>Workflow</i> o "Flujo de Trabajo"	53
5.5.4 Implementando el Workflow en Notes	55
5.5.5 Base de datos de Notes	56
5.5.6 Notes en Internet [CHAVEZ, 96]	58
5.6 TECNOLOGÍA ODBC	62
5.7 Servicios de ODBC API	63

5.8 Tipos de manejadores de ODBC	64
5.9 Niveles de conformación del ODBC API	64
5.10 Niveles de Conformación del ODBC SQL	65
6. ANÁLISIS DEL SISTEMA	66
6.1 Análisis de la Estructura de Objetos (AEO)	66
6.1.1 Identificación de Tipos de Objetos	67
6.1.2 Esquema de Asociaciones de Objetos	68
6.1.3 Jerarquías de Generalización de Objetos	69
6.2 Análisis del Comportamiento de Objetos (ACO)	70
6.2.1 Definición del enfoque del análisis	70
6.2.2 Aclaración del Tipo de Evento	71
6.2.3 Identificación de los estados permisibles	72
6.2.4 Definición de las condiciones de control	72
7. DISEÑO DEL SISTEMA	76
7.1 Diseño de la Estructura y el Comportamiento de Objetos a nivel Componentes	77
7.2 Diagramas de DEO y DCO de los Componentes del Sistema	78
7.2.1 Clase DOCUMENTO	78
7.2.2 Clase ADMINISTRADOR ACCESOS	80
7.2.3 Clase ACLARACIÓN	82
7.2.4 Clase RECIBO	83
7.2.5 Elemento RELOJ	84
7.2.6 Clase ADMINISTRADOR DE EVENTOS	85
7.3 Resumen del DEO / DCO para los componentes del sistema	88
7.4 Diseño de la Estructura y Comportamiento de Objetos a nivel Sistema	89
7.5 Diseño de la Base de Datos Relacional (BDR)	93
7.6 Diseño de la Base de Documentos	94
7.7 Diseño de las vistas	95
7.7.1 Vistas de Administración	96
7.7.2 Vistas de Tipo Personal	96
7.7.3 Vistas de Tipo RH	97
7.8 Diseño del Recibo	99
8. CONSTRUCCIÓN DEL SISTEMA	100
8.1 Arquitectura General del Sistema	100
8.2 Arquitectura bajo el concepto Cliente / Servidor	101

ÍNDICE

8.3 Instalación y Configuración del manejador RUMBA	104
8.3.1 Consideraciones antes de la configuración	105
8.3.2 Instalación del manejador	106
8.3.3 Configuración del manejador	106
9. Administración y Operación del Sistema	108
9.1 Documentos para la Administración del Sistema	108
9.2 Roles y niveles de acceso en la Administración del Sistema	109
9.3 Aspectos de la Operación del Sistema	111
9.4 Delimitación de Responsabilidades	113
10. CONCLUSIONES	100
10.1 Metodologías de desarrollo	114
10.2 Tecnologías empleadas	115
10.3 Plataforma de desarrollo	115
10.4 Conclusiones generales	116
11. BIBLIOGRAFÍA	117
11.1 BIBLIOGRAFÍA DE REFERENCIAS	117
11.2 BIBLIOGRAFÍA GENERAL	118

Indice de Tablas

<i>Tabla 3.1 Roles de los agentes que intervienen en el proceso de Verificación de Pago de Nómina</i>	15
<i>Tabla 5.1 Características de los sistemas de tiempo compartido y recursos compartidos"</i>	32
<i>Tabla 5.2 Gramática de SQL, tipos de datos y expresiones soportadas por los niveles de conformación de ODBC</i>	65
<i>Tabla 6.1 Aclaración del tipo de evento</i>	71
<i>Tabla 7.1 Atributos de la Clase Documento</i>	80
<i>Tabla 7.2 Métodos de la Clase Documento</i>	80
<i>Tabla 7.3 Atributos de la Clase Administración de Accesos</i>	81
<i>Tabla 7.4 Métodos de la Clase Administración de Accesos</i>	81
<i>Tabla 7.5 Atributos de la Clase Aclaración</i>	82
<i>Tabla 7.6 Métodos de la Clase Aclaración</i>	83
<i>Tabla 7.7 Atributos de la clase Recibo</i>	84
<i>Tabla 7.8 Métodos relacionados con la clase ADMINISTRADOR DE EVENTOS</i>	87
<i>Tabla 7.9 Resumen del DEO / DCO de los componentes del sistema</i>	88
<i>Tabla 7.10 Equivalencias aproximadas entre una base de datos y una base de documentos</i>	94

Indice de figuras

<i>Figura 3.1</i>	<i>Proceso Actual de Verificación de Pago de Nómina</i>	16
<i>Figura 3.2</i>	<i>Esquema de solución propuesto</i>	21
<i>Figura 5.1</i>	<i>Tipos de procesos distribuidos</i>	34
<i>Figura 5.2</i>	<i>Configuraciones de Sistemas Cliente / Servidor</i>	35
<i>Figura 5.3</i>	<i>Procesos Servidores Maestro/Esclavos</i>	37
<i>Figura 5.4</i>	<i>Comparación entre aplicaciones de BD Cliente/Servidor y Archivos Compartidos</i>	38
<i>Figura 5.5</i>	<i>Comunicación RPC</i>	39
<i>Figura 5.6</i>	<i>Modelación de la realidad empleando la Metodología Orientada a Objetos (MOO)</i>	47
<i>Figura 5.7</i>	<i>Modelos del análisis y diseño orientado a objetos</i>	48
<i>Figura 5.8</i>	<i>Arquitectura de funcionamiento de InterNotes Web Publisher</i>	59
<i>Figura 5.9</i>	<i>Arquitectura de funcionamiento del Domino</i>	61
<i>Figura 6.1</i>	<i>Esquema de Asociaciones de Objetos</i>	68
<i>Figura 6.2</i>	<i>Jerarquía de Generalización. Objeto Base de Documentos</i>	69
<i>Figura 6.3</i>	<i>Jerarquía de Generalización de dos tipos de objetos compuestos</i>	70
<i>Figura 6.4</i>	<i>Diagrama de transición válida de estados</i>	72
<i>Figura 6.5</i>	<i>Esquema de Eventos. a) Proceso de Importación de la Nómina y Notificación al Usuario</i>	74
<i>Figura 6.6</i>	<i>Esquema de Eventos. b) Proceso de Consulta, Aclaración, Firma e Impresión de Recibo</i>	74
<i>Figura 6.7</i>	<i>Esquema de Eventos. c) Proceso de revisión de Aclaraciones y Comentarios</i>	75
<i>Figura 6.8</i>	<i>Esquema de Eventos. d) Procesos Generador de Recordatorios y Verificación de Pendientes</i>	75
<i>Figura 7.1</i>	<i>Esquema de la Estructura de una clase</i>	77
<i>Figura 7.2</i>	<i>Diagrama DEO/DCO para la Clase Documento</i>	79
<i>Figura 7.3</i>	<i>Diagrama DEO/DCO para la Clase Administración de Accesos</i>	81
<i>Figura 7.4</i>	<i>Diagrama DEO/DCO para la Clase Aclaración</i>	82
<i>Figura 7.5</i>	<i>Diagrama DEO/DCO para la clase Recibo</i>	83
<i>Figura 7.6</i>	<i>Diagrama DEO/DCO para el Agente Reloj</i>	85
<i>Figura 7.7</i>	<i>Diagrama DEO / DCO para la clase ADMINISTRADOR DE EVENTOS</i>	86
<i>Figura 7.8</i>	<i>Estructura de Tablas de información de Nómina</i>	94
<i>Figura 7.9</i>	<i>Diseño de las vistas</i>	95
<i>Figura 7.10</i>	<i>Diseño de la vista Configuración del Sistema</i>	97
<i>Figura 7.11</i>	<i>Diseño de la vista Solicitudes de Aclaración</i>	98
<i>Figura 8.1</i>	<i>Arquitectura General del Sistema</i>	101
<i>Figura 8.2</i>	<i>Arquitectura del sistema desde el punto de vista de los procesos del usuario</i>	103
<i>Figura 8.3</i>	<i>Arquitectura del Sistema desde el punto de vista del servidor</i>	104
Figura 9.1	Ejemplo de un documento de Registro de Personal	109
Figura 9.2	Ejemplo del documento de Parametrización del Sistema	110
<i>Figura 9.3</i>	<i>Documento Mensaje de Nómina</i>	112
Figura 9.4	Documento Solicitud de Aclaración	112

Indice de Escenarios

<i>Escenario I Importación de la Nómina y notificación al usuario</i>	89
<i>Escenario II Consulta de la Información de Nómina</i>	90
<i>Escenario III Activación de Eventos</i>	91
<i>Escenario IV Revisión, envío y respuesta de aclaraciones</i>	92
<i>Escenario V Recordatorio Automático</i>	93
<i>Escenario VI Interacción de objetos con el exterior a través de sus métodos</i>	120

RESUMEN

El presente documento es el resultado de un proyecto realizado en Cemtec, S.A. de C.V. y presentado a la Universidad de Monterrey como Proceso de Evaluación Final. Este proyecto integra los conceptos Análisis y Diseño Orientado a Objetos, tecnología Cliente / Servidor, Workflow y acceso desde Internet para producir una herramienta que apoye el proceso de Verificación de Pago de Nómina y monitoreo de firmas electrónicas.

La plataforma en la que se desarrolló es Lotus Notes versión 4.5, conocida también como Domino, la cual entre sus características más sobresalientes se encuentra su funcionamiento como servidor de Web.

INTRODUCCIÓN

En un afán de globalización del mercado, las empresas constantemente se topan con nuevos retos tecnológicos destinados principalmente a acortar las distancias cada vez mayores entre las unidades de negocio. Uno de los retos más grandes consiste en mantener las diferentes unidades de negocio con una comunicación que no dependa del tiempo ni del espacio. Cemtec ha respondido a estas necesidades a través de su implementación muy particular del concepto “*Flujo de Trabajo*”, también conocido como *Workflow*, con lo que ha denominado una **Herramienta para la Coordinación de Acciones (HCA)**. Una HCA agiliza el proceso de toma de decisiones, apoya el seguimiento de un flujo de documentos y coordina cumplimiento de tareas en grupos dispersos en el tiempo y en el espacio a través de lo que se denomina “*conversaciones*” entre los participantes.

El presente documento es el reflejo de un semestre de trabajo en el desarrollo de una HCA que soporta las conversaciones involucradas en el proceso de Verificación del Pago de Nómina y monitoreo de firmas electrónicas.

En primera instancia, el lector conocerá a Cemtec como empresa y los servicios que provee a fin de situarlo en la naturaleza y dimensión de la empresa. El segundo capítulo está destinado a detallar el concepto Herramienta para la Coordinación de Acciones con el objetivo de que el lector conozca a grandes rasgos lo que significa una conversación en términos de Procesos del Negocio y los beneficios de su implementación tecnológica.

El *ESTUDIO PRELIMINAR* revela en el capítulo tres la situación problemática que se experimentó a través de una narrativa acompañada de ilustraciones que exponen la manera en que se realiza el proceso actual de Verificación de Pago de Nómina, así como su análoga solución propuesta mediante una descripción de las características del nuevo proceso; adicionalmente se presentan también en esta sección los requerimientos y características del producto a desarrollar, las ventajas que se esperan obtener de esta nueva implementación y las dificultades que se pueden presentar durante el desarrollo del proyecto.

Adentrados en la problemática y su propuesta de solución, el capítulo 4 denominado *DELIMITACIÓN DEL PROYECTO* pretende encerrar el alcance del proyecto en objetivos mensurables, con la finalidad de ser específico en cuanto a lo que hará o no hará la herramienta a desarrollar.

El *MARCO TEÓRICO* es una síntesis de los conceptos básicos en metodología y tecnología de desarrollo que fueron empleados en el desarrollo de este proyecto; el objetivo de esta sección es que el lector se familiarice con estos términos con la finalidad de que comprenda mejor los términos y conceptos utilizados en los capítulos subsiguientes.

El capítulo 6 trata del Análisis del Sistema propiamente, introduciendo en primer lugar el Análisis de la Estructura de Objetos, el cual identifica los tipos de objetos existentes en el sistema y sus asociaciones, y en segundo lugar el Análisis del Comportamiento de Objetos el cual se encarga de identificar los estados en los que puede estar un tipo de objeto dado y los eventos de objetos válidos en cada proceso. Sustentado en los resultados del Análisis, el Diseño del Sistema es presentado en el capítulo 7; dos enfoques están incluidos en el Diseño de la Estructura y Comportamiento de Objetos, el primero muestra el diseño de los componentes del sistema (las clases por separado) y el segundo presenta el cómo interactúan estos componentes. Complementando el Diseño

de Objetos se presentan también el Diseño de Base de Datos Relacional, el Diseño de la Base de Documentos y el Diseño de las vistas.

El capítulo 8 toca los temas relevantes a la construcción del sistema. Los diferentes enfoques de la arquitectura final del sistema son expuestos, así como los detalles de la Instalación y Configuración del manejador RUMBA[®] ODBC.

Los aspectos relevantes a la Administración y la Operación del Sistema son tocados en el capítulo 9, incluyendo la delimitación de las responsabilidades para la correcta operación.

Con el objetivo de enriquecer el desarrollo de futuros proyectos, el capítulo 10 *CONCLUSIONES* habla sobre los descubrimientos más relevantes durante el desarrollo del sistema, así como de algunas recomendaciones para proyectos que empleen uno o más conceptos de los aquí expuestos.

Por último, se ofrece al lector un conjunto de referencias bibliográficas en las cuales podrá profundizar en los conceptos expuestos en este documento.

1. CEMTEC¹

En este capítulo se presentan las características, servicios y productos que ofrece Cemtec; el objetivo es que el lector se familiarice con el giro de la empresa y sus necesidades.

CEMTEC, S.A. de C.V. es una compañía con el perfil internacional filial del grupo Cemex, especializada en integrar Tecnología en Información (TI) a la estrategia de negocio de sus clientes.

CEMTEC se ha fortalecido a través de los años y esto le ha permitido concretar su estrategia de expansión y crecimiento. Al día de hoy, la compañía ha implantado soluciones integrales de Tecnología de Información a diferentes clientes y en diferentes países.

¹ Extraído de un boletín emitido a la Prensa

Actualmente, Cemtec brinda servicios en México, España, USA, Venezuela, Panamá, Colombia, Honduras y República Dominicana.

1.1 Antecedentes

El grupo Cemex es uno de los principales productores mundiales de cemento, concreto y derivados. Es una organización que se caracteriza, entre otras cosas por la búsqueda continua en lograr la integración transparente de la tecnología en sus procesos de negocio.

CEMTEC se creó con el propósito de proveer servicios de Tecnología de Información de la más alta calidad al grupo Cementos Mexicanos.

Así, se dio a procesos de fortalecimiento y desincorporación hacia una empresa autosuficiente, el cual comenzó formalmente en agosto de 1993. CEMTEC fue formada con personal de amplia experiencia en el uso de TI, quienes hasta entonces habían colaborado activamente en la exitosa definición e implantación de Tecnología de Información dentro del grupo CEMEX.

La empresa inició operaciones con un número aproximado de 50 personas. Actualmente cuenta con más de 170 profesionistas especializados en diferentes procesos de negocio y plataformas tecnológicas.

1.2 Misión de CEMTEC

La Misión de la empresa es “Proveer servicios de Tecnología de Información de Clase Mundial enfocados a mejorar la productividad y los procesos de negocio de nuestros clientes”.

La empresa busca un balance entre los servicios al cliente, el desarrollo profesional de sus integrantes y la Innovación e investigación de nuevas ofertas para alcanzar y mantener la competitividad en el mercado.

1.3 CEMTEC y el Servicio al Cliente

CEMTEC es una empresa en la cual el principal reto es lograr la satisfacción total de sus clientes a través de los productos y servicios que ofrece. En

CEMTEC el cliente es la persona más importante y la empresa promueve la filosofía de escuchar y observar para de esta manera entender el fondo de las necesidades de sus clientes y ofrecer soluciones integrales que vayan de acuerdo a los objetivos del negocio del cliente. Así mismo, una característica esencial de la empresa es que busca formar alianzas con otros proveedores especializados de TI para de esta manera proveer a sus clientes las mejores alternativas tecnológicas.

CEMTEC promueve la Tecnología de Información como un vehículo que permite generar y mantener redes de colaboración en la empresa, rompiendo barreras organizacionales y apoyando la flexibilidad que las empresas requieren para poder competir en el mercado nacional e internacional.

CEMTEC es una empresa que cuenta con personal comprometido con el servicio al cliente, la creatividad, la innovación tecnológica, la calidad total y la apertura ante el mundo cambiante de la Tecnología de Información.

1.4 Logros de CEMTEC

- CEMTEC contribuyó en el posicionamiento del Grupo Cemex dentro de los 100 mejores usuarios globales de Tecnología de Información, mención otorgada por la revista Computer World en Mayo de 1995.
- CEMTEC brinda a sus clientes Tecnología de Información que apoya a los procesos de negocio de la empresa, contribuyendo por un lado a la mejora de los procesos a través del rediseño y por otro, proveyendo herramientas tecnológicas que ayudan a incrementar la productividad y la rápida toma de decisiones.
- La infraestructura que CEMTEC ha desarrollado para sus clientes en Lotus Notes, soporta más de 5500 usuarios y 120 servidores en 8 países. Así, también es pionero en el desarrollo de 250 aplicaciones de Lotus Notes.
- Su experiencia en LAN / WAN comprende la instalación de más de 60 ruteadores, 90 redes locales y 5500 nodos de red a nivel mundial.

- Ha instalado más de 45 conmutadores digitales en una red privada y cuenta con gran experiencia en sistemas de correo de voz.

- CEMTEC ha implementado exitosamente el Distributed Data Processing (DDP) de JDEdwards en más de 25 equipos AS/400 en 5 países.

- CEMTEC ofrece el módulo de Nómina totalmente compatible con JDEdwards adecuado para funcionar en México, Panamá, República Dominicana y Honduras.

- CEMTEC ha establecido alianzas con diferentes empresas líderes en su ramo, las cuales fortalecen la oferta de soluciones globales de la empresa:
 - ◆ Action Technologies Preferred Business Partner
 - ◆ Centigram Authorized Reseller
 - ◆ Harris Authorized Reseller
 - ◆ IBM Business Partner
 - ◆ JDEdwards Consulting Service Provider
 - ◆ Lotus Authorized Education Center
 - ◆ Lotus Premium Business Partner
 - ◆ Microsoft Solution Provider
 - ◆ Universidad de Monterrey

1.5 Productos y Servicios

Cemtec ofrece una gamma muy amplia de productos y servicios enfocados a soluciones globales de cliente; a continuación se presentan.

1.5.1 Sistemas de Información Ejecutiva

Sistemas de apoyo en la toma de decisiones. Cubriendo los servicios de análisis, desarrollo, implementación, capacitación y soporte.

1.5.2 Herramienta para la Coordinación de Acciones (Workflow)

Rediseño de Procesos de Negocio y construcción de herramientas para la Coordinación de Acciones bajo las plataformas Lotus Notes y SQL Server.

CEMTEC es Preferred Business Partner de Action Technologies, proveedor líder en la industria de Workflow. El engine de Workflow de ATI es la base de coordinación de las Herramientas para la Coordinación de Acción y de Automatización de Procesos de Negocio.

Ofrece consultoría tecnológica y de procesos de negocio, así como venta de productos y licencias.

1.5.3 Automatización de Procesos

Integración de Plataformas como RS/600, AS/400, DOS y Windows (NT). Sistemas expertos orientados a la automatización de procesos de producción y transmisión de datos. Desarrollo de aplicaciones enfocadas a la interacción con equipo de control, recolección y transmisión de datos.

1.5.4 JDEdwards

CEMTEC es *Consulting Service Provider* de JDEdwards & Company. Ofrece planeación, instalación, mantenimiento, capacitación y soporte, así como desarrollo de adecuaciones específicas para los módulos de Abasto, Finanzas, Recursos Humanos, Mantenimiento de Equipo y Ventas.

Se cuenta con la única Nómina complementaria a la Nómina JDEdwards para México, Panamá, República Dominicana y Honduras. La Nómina CEMTEC está desarrollada bajo los mismos estándares de JDEdwards y construida sobre la misma base de la Nómina JDEdwards original.

1.5.5 Groupware

CEMTEC es *Premium Business Partner* de Lotus Development Corporation. Ofrecemos planeación, instalación y mantenimiento de infraestructura Lotus Notes. Cuenta con experiencia en el desarrollo de más de 250 aplicaciones

institucionales, Planeación, Instalación y mantenimiento de companion products de Lotus Notes. Venta de productos y licencias de Lotus.

1.5.6 Internet

Estrategias enfocadas a utilizar en forma efectiva y productiva Internet en el negocio. Adicionalmente ofrece desarrollo de Sistemas de Información y aplicaciones con tecnología como HTML, Java y VRML, así como planeación e implementación de Intranets.

1.5.7 Networking

Diseño, planeación, instalación y diagnóstico de redes LAN/WAN.

1.5.8 Network & System Management

Planeación de Infraestructura de redes locales y amplias a nivel mundial, incluyendo servicios de consultoría, planeación, diseño, instalación y mantenimiento.

1.5.9 Telecomunicaciones

Implementación de telecomunicaciones a nivel mundial, comunicación de voz, datos y video por diferentes medios. Se cuenta con profesionistas certificados en Harris New y East (Sistemas de Satélite Intelast).

1.5.10 Telefonía

Distribuidores autorizados de Harris. Ofrece planeación, diseño e instalación de redes telefónicas utilizando conmutadores Harris. Experiencia en redes privadas. Instalación de sistemas de comunicación PCS.

1.5.11 Herramientas de apoyo Logístico

Consultoría, implementación y soporte sobre Herramientas de Localización Vehicular utilizando Sistemas de Información Geográfica y diferentes medios de

comunicación. Terminales a bordo para envío de estatus, datos y coordenadas GPS. Identificación Automática de Equipos.

1.5.12 Multimedia

Integración de tecnología Multimedia para generar un valor real en el negocio, enfocada a diversas aplicaciones como presentaciones corporativas, aplicaciones de publicidad y mercadotecnia, cursos interactivos y kioskos.

1.5.13 Centro de Educación

Cursos de la más alta calidad y adaptados a las necesidades particulares del negocio, calendario de cursos ya programados Windows, Windows 95, MS Office, Lotus Notes e Internet.

1.5.14 Help Desk

Asesoría y consultoría en el manejo de Aplicaciones de Oficina, permitiendo un uso más transparente de la tecnología e incrementando la productividad en el negocio. Servicio telefónico o vía electrónica.

2. HERRAMIENTA PARA LA COORDINACION DE ACCIONES (HCA)

En esta época donde la comunicación entre las personas es necesaria en las empresas para hacer más efectivas con sus procesos de negocios, es de suma importancia el facilitar esta comunicación y generar acciones a través de las conversaciones; lo que implícitamente facilitaría los procesos de negocios. Es la razón por la que a finales de 1993 Cemtec comenzó a recibir una transferencia de tecnología basada en una nueva manera de ver los procesos de negocios, estos procesos se basan en reconocer y coordinar los actos de habla involucrados.

En junio de 1994 se creó dentro de Cemtec el área funcional Herramientas para la Coordinación de Acciones, cuyo objetivo es el apoyar a las organizaciones a hacer más efectivos sus procesos de negocio haciendo uso del concepto "*Workflow*", pero no meramente el que implica mover papeles o materiales dentro de la empresa para reducir tiempos y movimientos, sino el que pretende mejorar la comunicación entre los participantes involucrados.

El concepto que aquí se maneja asume que cada persona de la organización puede actuar como cliente o como proveedor de cierta conversación. Un proceso de negocio es una red de conversaciones entre pares de clientes y proveedores, cuyo propósito es la completa satisfacción de las condiciones que ambos negocian y conocen.

Una HCA “**Herramienta de Coordinación de Acciones**” es una herramienta que apoya tecnológicamente las conversaciones que constituyen un Proceso de Negocio, teniendo como característica principal generar acciones y compromisos entre las conversaciones de los participantes, realizando los procesos de una forma más rápida, sencilla y clara para involucrarlos.

Las HCA's permiten:

1. Promover un lenguaje común entre los participantes del proceso.
2. Enfocar a los participantes a la satisfacción de sus clientes.
3. Monitorear el estatus de cada proceso.
4. Informar y recordar a los participantes del proceso sus compromisos pendientes.
5. Integrar las fuentes de información necesarias para apoyar el cumplimiento de los compromisos.
6. Reducir tiempos de ciclo del proceso automatizado.
7. Apoyar transparentemente el proceso sin importar barreras de tiempo y espacio.

Actualmente Cemtec ofrece servicios a clientes de talla de CEMEX donde se han instalado varias aplicaciones en sus plantas a través de todo el mundo.

Teniendo en cuenta el nivel de sus clientes y gracias a experiencias, Cemtec ha asimilado la necesidad de crear aplicaciones con un alto nivel de calidad, por lo que es de gran prioridad en el área contar con un equipo que esté constantemente en capacitación y certificación de las herramientas tecnológicas que utiliza para desarrollo y de esta forma estar siempre a la vanguardia para poder cumplir con las necesidades de competencia que sus clientes exigen. Tal es la importancia que se le da a este aspecto que el Ing. Omar Pechir (desarrollador y especialista en el ambiente Lotus Notes) es uno de los primeros en México que obtuvo la certificación por **Lotus Development Corporation como Certified Lotus Professional Application Developer (R4)** Desarrollador

profesional de Aplicaciones en Lotus Notes (v.4); lo cual es un gran logro tanto para la empresa como para sus clientes.

Dentro del área de HCA's se han desarrollado Herramientas que apoyan procesos administrativos, tales como:

HCA Gastos de Viaje (Lotus Notes)

Esta HCA apoya la administración del proceso requerido en un viaje de negocios, desde la solicitud de anticipos hasta el reporte de la cuenta de gastos y las actividades realizadas durante el viaje.

Rompe con la barrera del espacio y tiempo ya que es factible utilizarla desde cualquier lugar contando solamente con una computadora y un módem.

Proporciona un control estadístico de fácil consulta y actualización por cada viaje realizado.

HCA Pedidos y Ofertas (Lotus Notes)

Apoya efectivamente el seguimiento de las conversaciones que se dan en torno a un pedido o a una oferta entre grupos de trabajos.

HCA Pago Electrónico (VB-SQL)

Esta HCA apoya el proceso de entrega de los recibos de nómina y de su firma electrónica, en forma rápida y eficiente.

Además del desarrollo de Herramientas para la Coordinación de Acciones, dentro del área Cemtec ofrece asesoría y soporte para la creación de nuevas herramientas que ayuden a sus clientes a mejorar sus procesos de negocios.

3. ESTUDIO PRELIMINAR

Este capítulo tiene como objetivo situar al lector dentro de la problemática experimentada y cómo ésta presuntamente será resuelta. Contiene una descripción del proceso actual de Verificación de Pago de Nómina y una propuesta de solución describiendo las ventajas de su implementación y aspectos como características y requerimientos relacionados con el producto final que se pretende obtener.

3.1 Antecedentes y descripción del problema

El sistema de Nómina constituye parte esencial del conjunto de herramientas de informática en el que se apoya casi cualquier empresa de tamaño mediano a grande. La necesidad de informar a los empleados el monto de sus percepciones y deducciones implicados por la prestación de sus servicios a la empresa se hace presente entonces. CEMTEC respondió a esta necesidad con el desarrollo de una versión temprana de la llamada "HCA² Pago de Nómina". Esta herramienta se encuentra actualmente implementada en

² HCA significa de aquí en adelante en este documento: Herramienta para la Coordinación de Acciones.

CEMTEC en el servidor SOCMT01 y en el SOCMT11, así como en el Corporativo de Cementos Mexicanos, y es un desarrollo del equipo especializado en HCA's.

La primera versión de la HCA de Pago de Nómina una vez que se ha ejecutado el proceso de Nómina y mediante un conjunto de operaciones efectuadas (desde y fuera de la herramienta) por el área de Recursos Humanos de la empresa, pretende informar a los empleados el monto de las percepciones y deducciones del mes y los conceptos correspondientes (sea por concepto de sueldo o pago especial³). Existen tres agentes que intervienen en este proceso: 1) El administrador de Recursos Humanos, 2) el empleado y 3) Observadores generales; cada uno tiene su propio rol dentro de la aplicación (como se muestra en la Tabla 3.1), formando lo que CEMTEC llama *conversaciones*⁴ entre ambos agentes.

ROL	FUNCION / DESCRIPCION
Asesor de Recursos Humanos	Entre sus funciones se encuentran: <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Importar la información del pago de nómina a través de la HCA. <input checked="" type="checkbox"/> Atender a las sugerencias o comentarios y completar el ciclo de aclaraciones sobre el pago.
Empleado	Una vez que revisa el pago correspondiente al último ciclo de nómina, o algún pago especial puede tomar alguna de las siguientes acciones: <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Firmar el recibo (electrónicamente). <input checked="" type="checkbox"/> Solicitar aclaración
Observador	Tiene acceso a consultar la información de pago de todos los empleados a su cargo. Comúnmente será un gerente o un responsable de área.

Tabla 3.1 Roles de los agentes que intervienen en el proceso de Verificación de Pago de Nómina

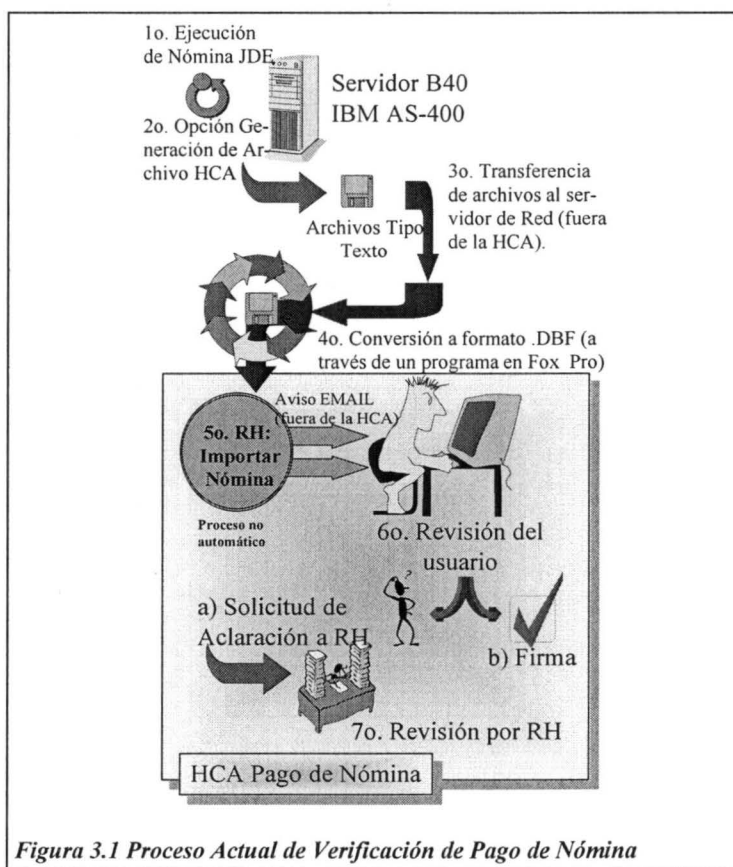
³ Un pago especial puede ser percepciones extraordinarias como el Aguinaldo, el Reparto de Utilidades o mecanismos como el Fondo de Ahorro.

⁴ Una conversación es un proceso en el que existe un ente (o un conjunto de ellos) que lo inicia a través de una petición y otro(s) responde a ella con una acción o una serie de ellas.

3.1.1 Sobre el proceso actual

La Figura 3.1 muestra cómo se lleva a cabo el proceso de Verificación de Pago de Nómina en la actualidad:

1. **Ejecución de Nómina JDE.** En CEMTEC, en el Corporativo de CEMEX y en casi todas sus plantas en el mundo se ejecuta la Nómina de JDEdwards⁵, adaptada al sistema mexicano por los especialistas de CEMTEC.



⁵ JDEdwards (abreviado como JDE) es una compañía estadounidense cuyos sistemas globales de Contabilidad, Nómina, Abasto, Mantenimiento y muchos otros son reconocidos mundialmente. CEMTEC tiene el reconocimiento de JDEdwards como Business Partner.

2. **Opción de *Generación de Archivo HCA***. La adaptación de la Nómina JDEdwards para México tiene una opción adicional al proceso (desarrollada en CEMTEC) que es ejecutada hacia el final por el mismo operador de la Nómina. Esta opción genera un conjunto de archivos tipo texto que contienen la información del último pago de nómina. Esta información está constituida por:
 - Periodos
 - Conceptos
 - Depósitos (Montos totales)
 - Importes (Desglose por concepto)
 - Fondo de Ahorro
 - Saldos
 - Nombres de los Trabajadores
3. **Transferencia de archivos al servidor de Red**. Cada uno de los archivos arriba citados son enviados uno a uno desde el AS-400 hacia el servidor de Red Novell llamado SOCMT11. Este proceso es efectuado por el personal de Recursos Humanos de la empresa y lo realiza por medio de un conjunto de programas de transferencia ejecutados desde una terminal con Windows 95.
4. **Conversión a formato .DBF** Los archivos localizados en el servidor de Red Novell son tomados por un programa desarrollado en Fox Pro 2.5, traducidos en un formato legible por SQL y dispuestos en un directorio en particular en el mismo servidor de Red. Cabe recalcar que tanto la transferencia de archivos al servidor de Red y la conversión a formato .DBF son procesos que no se realizan en forma automática, sino que son efectuadas por el administrador de Recursos Humanos con programas auxiliares (fuera de la HCA) que fueron construidos para este propósito.
5. **HCA: Opción *Importar Nómina***. Una vez cubiertos los pasos anteriores, el administrador de Recursos Humanos encargado accesa esta opción en la HCA de Verificación de Pago de Nómina para que la base de datos en el formato .DBF sea convertida a una base de datos en SQL Server que será depositada posteriormente en el servidor de red SOCTM03 que funge como servidor de SQL. Este proceso aunque es periódico, no se realiza en forma automática.
6. **Revisión por parte del usuario**. El administrador de Recursos Humanos envía un mensaje a través del correo electrónico (fuera de la herramienta) para avisar a los usuarios que ya pueden consultar su pago correspondiente al último mes. El empleado (usuario) revisa los

conceptos de percepciones y deducciones de su pago; entonces puede ocurrir una de las siguientes acciones:

- a) **Solicitud de Aclaración.** En el caso de que la información consultada en la herramienta se encuentre incorrecta, incompleta o confusa, el empleado puede solicitar una aclaración al departamento de Recursos Humanos. Esta aclaración se deposita en la base de datos de la herramienta como un texto en donde el empleado explica el por qué de la solicitud y genera un aviso a través de correo electrónico al departamento de Recursos Humanos.
- b) **Emisión de Firma.** Si el empleado está de acuerdo con la información que consultó en la herramienta emite una firma electrónica a través de la misma, la cual es equivalente a la firma física de un recibo.

7. **Revisión por parte de Recursos Humanos.** El encargado de Recursos Humanos accesa la herramienta para efectuar una revisión de los comentarios y sugerencias en torno a la verificación del pago de nómina. Este proceso se efectúa en un periodo no fijo, es decir, la revisión se realiza a voluntad del encargado de Recursos Humanos.

3.1.2 Identificación del Problema

La implementación actual de Verificación de Pago de Nómina es una aplicación escrita bajo la plataforma de Visual Basic / SQL. Esta tecnología impide ciertas funcionalidades que son intrínsecas a las herramientas de la corriente de *Groupware*, como el caso de Lotus Notes (estándar de *Groupware*), tales como:

1. Exceso de operaciones intermedias entre la ejecución del proceso de Nómina y la ejecución de la Herramienta que dificultan y entorpecen el proceso. Estas operaciones incluyen la Generación de Archivo HCA, la transferencia de archivos al servidor de red, la conversión al formato .DBF y la Importación al HCA (pasos 2 al 5 en la Figura 3.1), las cuales resultan bromosas, imprácticas e inconvenientes si se desea comercializar este producto.
2. Esfuerzo adicional en la instalación. Cada PC que deba contar con esta aplicación requiere de la instalación de un conjunto de archivos tipo DLL para su ejecución.
3. Un cambio en el diseño de la aplicación implica la instalación de la misma en cada uno de los servidores que se requiera, además de la instalación de los archivos DLL correspondientes.

4. Debido a la naturaleza de la plataforma de la herramienta que se utiliza actualmente (VB/SQL), no es posible la integración de la funcionalidad del correo electrónico en un 100% al proceso; el administrador de Recursos Humanos no puede enviar un mensaje de notificación a través de la herramienta una vez que la nómina ya está depositada y no puede consultar directamente los comentarios o sugerencias de los empleados, sino que tiene que hacerlo forzosamente por medio de la herramienta.

3.2 Solución propuesta al problema

Se propone el desarrollo de una nueva versión de la HCA de Verificación de Pago de Nómina sobre la plataforma de Lotus Notes 4.5 que conserve la funcionalidad esencial descrita en la sección anterior pero que aproveche las ventajas de la infraestructura de informática de una intranet añadiendo algunas nuevas características. Se sugiere el empleo de análisis y diseño orientado a objetos por la naturaleza de la plataforma [MARTIN,94].

La plataforma Lotus Notes 4.5 conocida como Domino 4.5, es la respuesta de Lotus Notes a corporaciones como Microsoft y Netscape en cuanto a Internet se refiere. Basta con instalar Domino en un server, y éste fungirá como Web Server de igual manera como lo haría el Server de Netscape. El beneficio que se obtendrá consistirá en que tan sólo bastará con instalar Domino en el servidor (que fungirá como Web Server) para poder acceder la HCA desde cualquier *browser* que esté conectado a la intranet; lo cual no impide que la HCA también pueda ser accesada por medio de un cliente Lotus Notes.

Se sugiere además establecer la comunicación entre Windows NT (del server de Domino) y el IBM AS/400 (en donde se ejecuta la Nómina JDE) a través de la tecnología ODBC⁶ [JENNINGS, 94], además de plantear la alternativa al usuario de proveer la información del pago de nómina a través de un grupo de archivos tipo texto dispuestos en un servidor que sea direccionable directamente desde el servidor de Windows NT (que puede ser un servidor de Novell, por ejemplo).

⁶ ODBC (Open Database Connectivity) establece el estándar de conectividad entre bases de datos relacionales de diferentes plataformas.

3.2.1 Características del proceso

El esquema de solución propuesto pretende eliminar algunas de las operaciones intermedias existentes en el proceso actual (Figura 3.1). Este esquema es representado gráficamente en la Figura 3.2 y descrito a continuación. Los roles de los agentes que intervienen en este *proceso* (ver la sección *Antecedentes y descripción del problema*, pág #14) permanecen sin cambio.

1. **Ejecución de la Nómina.** El proceso de Nómina en sí permanece intacto, ya que está fuera del alcance de este proyecto; sin embargo se asumirá que el producto de éste será el siguiente⁷:

- Archivo Tipo Texto. El cual estará dispuesto en una dirección de un servidor accesible directamente desde el servidor de Windows NT donde radica la HCA. Además, este archivo deberá consistir en la información de pago de nómina en un formato preestablecido (durante el desarrollo del PEF).

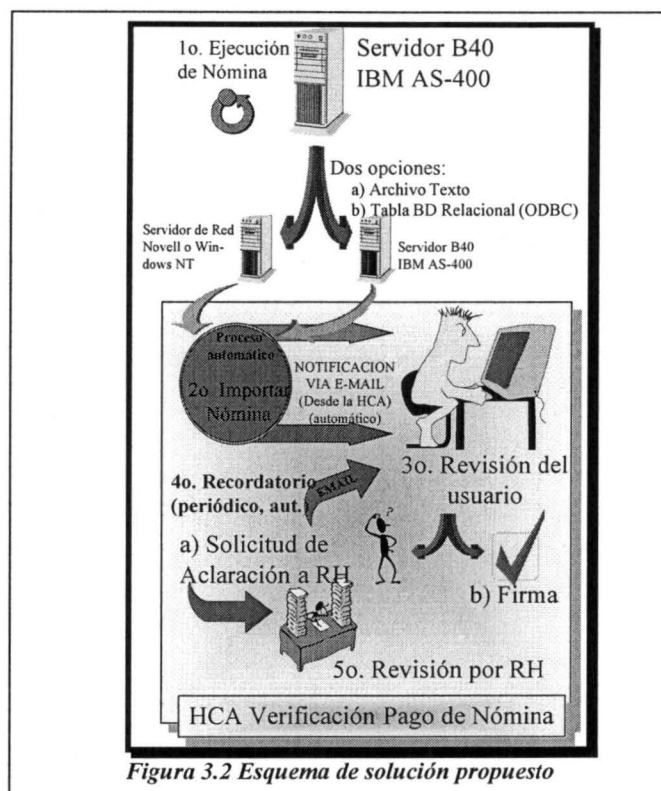
- Tabla BD Relacional. Como se explicó anteriormente, se empleará la tecnología ODBC para establecer la comunicación entre el servidor NT donde radica la HCA y el IBM AS-400 donde se ejecuta la nómina de manera que el proceso de importación de nómina será transparente para el servidor de Domino, y directo de su plataforma origen⁸. Se definirá también un formato estándar para la tabla que generará el driver de ODBC.

2. **HCA: Opción Importar Nómina.** Este proceso será periódico y automático (configurable). Esta opción toma la información del pago de la nómina para depositarlos en la base de datos de la herramienta. La importación puede ser a partir del archivo tipo texto o de la tabla generada por el driver de ODBC desde la plataforma donde se ejecutó la Nómina. El producto de la importación será una tabla en la base de datos de Lotus Notes que contendrá la información del pago de Nómina para que pueda ser interpretada fácilmente por la herramienta. Al final de la importación la herramienta generará automáticamente un aviso personal a través del correo electrónico a cada uno de los empleados

⁷ La idea original es que un sistema de nómina arroje uno de los dos productos, el que más le convenga al cliente que adquiera la HCA. Para efectos del PEF, se evaluarán ambos.

⁸ En teoría, el empleo de la tecnología ODBC permite la conectividad entre casi cualquier plataforma con un mínimo de ajustes (lo cual representa una ventaja sustancial para CEMTEC), sin embargo está fuera del alcance de este proyecto demostrar esto.

datos de alta en la lista de control de accesos de la herramienta notificando que su nómina ya está depositada y que pueden consultar esto en la HCA.



3. **Revisión por parte del usuario.** La revisión del usuario es parte de la definición de las conversaciones entre el empleado y el departamento de Recursos Humanos. La esencia de este proceso seguirá siendo la misma, el empleado verificará que los datos que aparecen en la herramienta estén correctos y realizará los pasos “a” o “b”.
 - a) **Emisión de Firma.** Una vez más si el empleado está de acuerdo con la información que consultó en la herramienta, emitirá una firma electrónica que se registrará en la base de datos de la misma.
 - b) **Solicitud de Aclaración.** En el caso de que la información consultada en la herramienta se encuentre incorrecta, incompleta o confusa, el empleado podrá solicitar una aclaración al departamento de Recursos Humanos a través de correo electrónico Lotus Notes (dentro de la herramienta) explicando en un texto el por qué de la solicitud.
4. **Recordatorio periódico y automático.** El proceso de recordatorio se realizará periódicamente en un lapso que será configurable. Se pretende que la herramienta genere los recordatorios en forma automática cuando existan empleados que tengan recibos pendientes de firmar.

5. **Revisión por parte de Recursos Humanos.** En caso de existir una aclaración, el encargado de Recursos Humanos recibirá una notificación vía correo electrónico de que existe un comentario o sugerencia referente a la consulta del pago de nómina y desde ahí mismo puede proceder a la revisión del mismo. Este proceso se podrá efectuar tanto en forma interactiva como en la forma tradicional (consultando la base de datos de la herramienta).

Como puede observarse, se ha rediseñado la parte del proceso que se refiere a la transferencia de la información que generó el proceso de Nómina hacia la herramienta, de manera de que se requiera una mínima intervención humana.

3.2.2 Requerimientos de la herramienta

Se desarrollará una herramienta en ambiente Intranet con una base de datos en Lotus Notes que pueda ser accesada desde cualquier terminal inteligente PC que esté conectada a la red y que cuente con un navegador de Internet *Netscape* o *Internet Explorer*, o en su defecto un cliente de *Lotus Notes V4.5* o posterior.

Los requerimientos de hardware de la terminal son:

MINIMO	RECOMENDABLE
Procesador 486 / 66 Mhz	Procesador Pentium 166 Mhz
RAM 8 Mb	RAM 16-32 Mb
Disco duro 30 Mb (disponibles)	Disco duro 30 Mb o más (disponibles)
Protocolo de com. TCP / IP	Protocolo de com. TCP / IP

Una terminal con acceso a la herramienta podrá estar conectada a una WAN, LAN o intranet que se encuentre ligada a un servidor de Lotus Notes V4.5 o posterior, o bien podrá trabajar en forma independiente mediante una creación

de una réplica de la base de datos. Esta réplica representa una fotografía de los registros del usuario en la base de datos de la herramienta

3.2.3 Características de la herramienta

La herramienta deberá soportar las conversaciones entre el personal empleado y el departamento de Recursos Humanos en lo referente a la verificación del pago de nómina (tal y como se indicó en la Figura 3.2). Entre las funcionalidades adicionales que deberá tener la herramienta se encuentran:

- ✓ **Integración del correo electrónico en un 100% al proceso.** La nueva herramienta será capaz de:
 - ☒ Notificar a los empleados automáticamente que su nómina ya fue depositada y que pueden consultar el desglose en cualquier momento.
 - ☒ Enviar a Recursos Humanos a través de correo electrónico las dudas o aclaraciones que tengan los empleados después de consultar el desglose de su pago.
 - ☒ Contestar las dudas o aclaraciones de los empleados (por parte de Recursos Humanos).
 - ☒ Manejo de recordatorios. Existirá un proceso periódico que emitirá mensajes vía correo electrónico para recordar a los empleados que tengan pendiente de declarar alguna satisfacción (o firma electrónica).
- ✓ **Apoyo total a las conversaciones del proceso.** Debido a que la herramienta se desarrollará en Lotus Notes, las funcionalidades intrínsecas de la corriente de *Groupware* se integrarán de forma natural al proceso. Estas funcionalidades son:
 - ☒ **Replicación.** Esta funcionalidad permite en un momento dado sacar una fotografía (copia) de la información de los registros del empleado para depositarlos en una réplica de la base de datos, con el objetivo de acelerar el tiempo de acceso de acuerdo a la ubicación física de los usuarios⁹.
 - ☒ **Base de datos compartida.** Esta funcionalidad provee la capacidad de compartir la información de los recibos a todo el grupo de Recursos Humanos, de tal manera que puedan observar los compromisos de todos los empleados y de ellos mismos en cualquier momento y sin tener que estar físicamente reunidos.

⁹ Esta funcionalidad sólo trabaja en una terminal con software cliente de Lotus Notes.

- ✓ **Implementación de la herramienta bajo el concepto de cliente-universal.** El concepto de *cliente universal* consiste en tener a disposición un conjunto de herramientas sin tener que estar ligado completamente a un software específico como Lotus Notes, en este caso. De esta manera, con la nueva herramienta la información referente al pago de nómina podrá ser accesible tanto desde un *Browser* de Internet como desde un *cliente Lotus Notes* (siempre y cuando exista conexión a la Intranet¹⁰).
- ✓ **Base de datos en Lotus Notes.** Esto permitirá que la administración de la información sea más natural a la herramienta.
- ✓ **Accesible prácticamente desde cualquier lugar del mundo**
- ✓ **Emisión de recibo impreso.** Elaborará el recibo impreso en un formato que permita colocar los sellos y firmas correspondientes.

3.2.4 Ventajas que se pretende obtener

La razón principal de la importancia de este proyecto radica en las ventajas que CEMTEC pretende obtener al emigrar a una nueva plataforma, más que en el conjunto de funcionalidades adicionales que se esperan de la nueva herramienta. Entre las principales ventajas se encuentran:

- ✓ Accesibilidad desde cualquier lugar del mundo, al poder hacerse vía módem/internet, sin necesidad de estar conectado a una WAN. Además, si no se cuenta con un módem es posible realizar una réplica de la base de datos para trabajar en forma independiente de la misma manera que cuando se está conectado al servidor.
- ✓ Ahorro en mantenimiento de nuevas versiones así como en soporte a la operación. Esta ventaja radica en que un cambio en el diseño de la herramienta es transparente completamente para el cliente o cualquier otro servidor que la contenga. Los cambios son actualizados automáticamente a través de los servidores de Lotus Notes por medio de un comando de replicación. Con la herramienta que existe actualmente implementada, un cambio en el diseño implica la instalación de archivos tipo DLL en cada cliente de la red.

¹⁰ Opcionalmente la herramienta podrá publicar la información al exterior (internet), sin embargo por razones de seguridad se limitará exclusivamente a la Intranet.

- ✓ Permitir mantener la conversación de vender el producto a externos con la ventaja de fácil instalación/configuración/accesibilidad/soporte.
- ✓ Proporcionar un valor en cambio de prácticas que se traduce en un cambio de cultura de los participantes del proceso al administrar sus compromisos en la HCA de acuerdo a las políticas del proceso (v gr. la declaración de la satisfacción a través de la confirmación electrónica del recibo por parte de los empleados en un periodo determinado).
- ✓ Proporcionar un valor simbólico (en mejora de imagen) a las empresas que adquieran esta HCA (una empresa que se jacte de que sus procesos sean “paper less” podrá adquirir esta herramienta para mejorar su imagen ante sus propios clientes).

3.2.5 Dificultades que se pueden presentar

La versión 4.5 de Lotus Notes trae como funcionalidad principal la conversión de formas de Lotus Notes en código HTML. Se ha detectado que las acciones en el *script* (o lenguaje nativo) de Lotus Notes no son todas traducibles por el software, por lo tanto será necesario intervenir con programación directa en HTML, y con la utilización de algunos controles en ActiveX [Zimmerman, 96].

4. DELIMITACIÓN DEL PROYECTO

El presente capítulo se elaboró con la finalidad de establecer los límites bajo los cuales se construirá el producto. Se han especificado el Objetivo General del Proyecto, un conjunto de Objetivos Específicos que describen en términos de compromisos las características que deberá tener el sistema y finalmente una lista de los recursos con los que se deberá contar para que el proyecto culmine con éxito.

4.1 Objetivo General del Proyecto

Desarrollar una aplicación que facilite el flujo de la información relacionada con la verificación del pago de la Nómina, entre el personal empleado y el departamento de Recursos Humanos, integrando las características típicas de la tecnología Groupware, como lo son el correo electrónico, replicación de bases de datos, administración de pendientes y conversaciones entre los participantes que se traducirán en un cambio en la cultura organizacional de quien adquiera esta herramienta.

4.2 Alcance

La característica principal de este producto radicará en que las consultas de la información del pago de la nómina no estarán limitadas a ser accesadas a través de una plataforma específica (como Lotus Notes, por ejemplo) es decir, la herramienta podrá ser accesada tanto a través de una terminal cliente de Lotus Notes, como por medio del *browser* de internet **Netscape** y el **Internet Explorer** por cualquier empleado de la empresa registrado en la lista de control de acceso de la herramienta utilizando las facilidades de la red interna o bien por intranet.

Otra característica será la conectividad con la plataforma AS-400, permitiendo así la interacción con el sistema de Nómina JDEdwards en un proceso simplificado con la mínima interacción humana. Esta interacción se efectuará por medio de un archivo tipo texto y por medio de una tabla relacional (en la práctica se utiliza sólo uno de los dos).

4.3 Objetivos Particulares

Se han dividido los Objetivos Particulares en dos grupos; el primero de ellos expresa los compromisos desde el punto de vista del programador, el segundo grupo enumera los objetivos desde el punto de vista del producto final del desarrollo.

4.3.1 Objetivos personales

1. Capacitación en desarrollo Lotus Notes [LOTUS,97]
2. Aprender lenguaje HTML [Graham, 95].
3. Investigar factibilidad controles ActiveX o JAVA [Zimmerman, 96] / [Capione, 96].
4. Aprender análisis y diseño orientado a objetos [MARTIN, 94]
5. Desarrollar el análisis y diseño del sistema [SENN,92]
6. Integrar a la construcción de los módulos del sistema cada uno de los procesos identificados de acuerdo a la funcionalidad descrita en la sección objetivos del sistema:
 - 6.1. Correo electrónico
 - 6.2. Verificación de Pago de Nómina
 - 6.3. Información Histórica
 - 6.4. Verificación de pendientes
 - 6.5. Impresión de recibos

- 6.6. Interfaz con plataforma
- 6.7. Convertidor a Browser
- 6.8. Seguridad de Acceso
7. Probar e instalar el producto [PRESSMAN,93].
8. Diseñar una interfaz con el usuario, tanto para el que accesa a través de Lotus Notes, como para el que accesa por medio de un *browser* [DUMAS,88].
9. Revisar con el usuario y el asesor el avance del proyecto.
10. Preparar y presentar periódicamente los avances del proyecto.
11. Realizar el documento del Programa de Evaluación Final.

4.3.2 Objetivos del Sistema

1. Integrar 100% el uso del correo electrónico al proceso [POMPILI,95].
2. Proveer un medio de comunicación formal entre Recursos Humanos y los empleados tal y como se indica en la sección 3.2.3 de este documento.
3. Proveer a los empleados la consulta del saldo y desglose de su último pago de nómina, anteriores o cualquier otro concepto como Reparto de Utilidades, Fondo de Ahorro, Aguinaldo, Bonos de desempeño, etc. así como información histórica de los mismos.
4. El acceso podrá por cualquiera de las siguientes dos formas:
 - 4.1. A través de una terminal cliente de Lotus Notes 4.5 [LOTUS,96]
 - 4.2. Por medio de un browser de Internet NetScape o Internet Explorer conectado a la Intranet.
5. La seguridad de acceso a la herramienta se otorgará mediante la asignación de passwords¹¹, tanto en la terminal cliente de Lotus Notes como desde un browser.
6. Para el caso del acceso desde una terminal que tiene instalado el cliente, el HCA reconocerá el Identificador de usuario y el password que administra Lotus Notes.
7. Si el acceso se intenta a través de un browser, se desarrollará un módulo especial para verificar la consistencia del identificador del usuario y su password con los que administra Lotus Notes.
8. También de desarrollará un módulo especial para la administración de usuarios en donde se indicarán los accesos pertinentes y los roles de los agentes del proceso.
9. Se construirá un módulo para la obtención de la información de nómina de 2 maneras:

¹¹ Se refiere a una contraseña o código de acceso

- 9.1. Desde un archivo tipo Texto dispuesto en un servidor de red, para el caso en que el cliente así le convenga.
- 9.2. Desde un archivo generado por la nómina en la plataforma en la que ésta fue ejecutada.
10. Proveer comunicación con la plataforma AS-400 a través de la tecnología ODBC [WINDOWS95,97]..
11. El sistema será capaz de reconocer al menos tres perfiles: 1) el administrador de Recursos Humanos, 2) el empleado común y 3) el observador general, otorgando así los accesos pertinentes.
12. Se permitirá que el administrador de Recursos Humanos pueda disparar de manera manual el proceso de importación de la información en cualquier momento.
13. Desarrollar una Base de Documentos en el formato Lotus Notes [MARTIN,87] que deberá almacenar información histórica dentro de un periodo configurable. El periodo configurable lo establece el usuario según las políticas propias del proceso en la empresa cliente. Significa que si el periodo es de 6 meses, la base de datos contendrá siempre (en forma automática) la información histórica de los últimos 6 meses.
14. Generar como reporte la impresión del recibo de nómina con un formato que facilite las firmas y sellos que le darán validez oficial para su uso externo.
15. Apoyar el cumplimiento del compromiso del departamento de Recursos Humanos de entregar los recibos, y del compromiso del resto del personal para la declaración de satisfacción (o firma electrónica) de los mismos.
16. Administrar recordatorios automáticos para los participantes con compromisos atrasados (al personal que tiene firmas de recibo pendientes).









4.4 Recursos necesarios

4.4.1 Hardware

Cantidad	Descripción
1	Servidor IBM PC Server 320
1	PC Pentium, 100 Mhz, 16 Mb RAM, 1.2 Gb disco duro.

	Cableado DB9 o RJ45
1	Tarjeta Token Ring
1	Conexión TCP/IP

4.4.2 Software

-  Windows NT para el Servidor de Lotus Notes
-  Lotus Notes 4.5 client
-  Lotus Notes 4.5 server
-  MS Explorer y Netscape
-  MS Power Point 7.0 para las presentaciones.
-  MS Word 7.0 para la documentación
-  ABC Flow Charter 3.0 para la elaboración de los diagramas de flujo.
-  MS Project para el control del avance del proyecto y Gráfica de Gantt.

4.4.3 Literatura

Se requerirán todos los manuales de Lotus Notes para la versión 4.5, además de la bibliografía mencionada en este mismo documento, así como la ayuda en línea proporcionada por Lotus Notes.

4.4.4 Tiempo

El proyecto completo requiere de 700 horas hombre, pero para efectos del Proceso de Evaluación Final se limitará a 16 semanas con los requerimientos que se manifiestan en este documento.

5. MARCO TEÓRICO

La presente sección tiene como objetivo que el lector se familiarice con los términos y metodología empleados con la finalidad de que comprenda mejor lo que en este documento se expresa. Para una mayor información consúltese las referencias contenidas en cada uno de los temas que a continuación se presentan.

5.1 *TECNOLOGÍA CLIENTE / SERVIDOR*

La tecnología Cliente / Servidor es un concepto relativamente nuevo, y se constituye como producto de la evolución de la industria del procesamiento de datos. Esta sección presenta los conceptos más importantes que apoyan al Objetivo Personal No. 6 y el Objetivo del Sistema No. 7, principalmente.

La tecnología cliente / servidor combina las mejores características de los sistemas de tiempo compartido tradicionales y los sistemas de recursos compartidos [Contreras, 94] (conocidos como redes de área local) tan ampliamente usados hoy en día (Tabla 5.1). En la implementación, la

arquitectura cliente / servidor se apoya en dos componentes fundamentales para combinar estas características.

Sistemas de tiempo compartido	Sistemas de recursos compartidos
<ul style="list-style-type: none"> ✓ Computadora central con alto poder ✓ Seguridad y administración de los datos ✓ Información centralizada, por lo tanto mayor seguridad en la integridad de los datos ✓ Administración de datos centralizada ✓ Capacidad para atender a muchos usuarios ✓ Terminales tontas ✓ Interfaz compleja y nada amigable ✓ Costo inicial muy alto 	<ul style="list-style-type: none"> ✓ Ampliamente utilizados en la actualidad ✓ Alta velocidad a un bajo costo ✓ Compartición de recursos ✓ Compartición de información ✓ Mayor cantidad de usuarios ✓ Propicio para corriente Groupware y Workflow ✓ Múltiples servidores administrando recursos ✓ Interfaz gráfica y amigable ✓ Flexibilidad en la implementación (muchos protocolos) ✓ No garantizan la integridad de la información

Tabla 5.1 Características de los sistemas de tiempo compartido y recursos compartidos"

El primero de estos componentes es el llamado *front-end*. El *front-end* se encarga de todo lo referente a la interfaz con el usuario, como las entradas estándar, texto, gráficas y las animaciones. Este elemento se ha visto fortalecido por las actuales modelos de desarrollo que permiten desarrollar cada vez interfaces más complejas en un tiempo despreciable (consúltese la sección Modelo Orientado a Objetos).

El segundo componente es llamado *back-end*. Este componente realiza la administración de los recursos centrales y el procesamiento de los datos.

El *front-end*, dispuesto normalmente en una estación de trabajo, se comunica con el *back-end*, que normalmente es un servidor a través de una red. Entonces puede asumirse que podemos tener muchos *front-end* para cada *back-end* en la red, y a la vez varios *back-end* atendiendo diferentes necesidades de los *front-end*.

La idea de separar el poder computacional en dos componentes es disminuir la carga de procesamiento del servidor central que atiende a múltiples usuarios, para llevarla a las diferentes estaciones de trabajo que suponen atender a un sólo usuario. Cabe señalar que este modelo no impide toda la funcionalidad de una computadora personal dentro de una red de área local (LAN, por sus siglas en inglés).

5.1.1 Conceptos

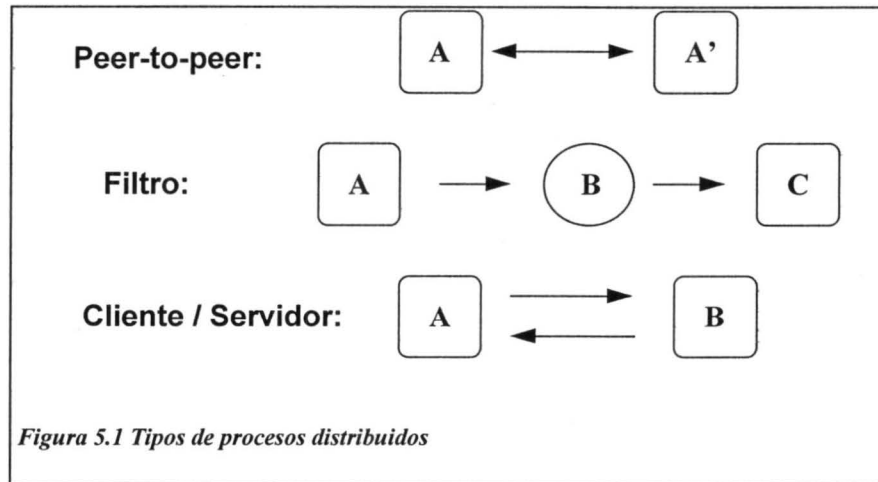
Antes de comenzar con lo que esta tecnología afirma que “debe ser”, hay que aclarar que cliente / servidor es un paradigma, y que si bien está compuesto por dos piezas (un cliente y un servidor), éstas pueden o no estar físicamente distantes. Lo importante es señalar que es un modelo para la interacción entre la ejecución de procesos concurrentes.

Los expertos no se han puesto de acuerdo en una definición; algunos de ellos definen los sistemas cliente / servidor como cualquier aplicación en la cual el solicitante de una acción se encuentra en un sistema y el proveedor de la acción está potencialmente en otro [Contreras, 93].

Cuando hablamos de que existe un solicitante de una acción y un proveedor, debemos asumir que existe comunicación entre esos procesos; esta comunicación en cualquier sistema multitarea se da como un proceso concurrente. Los procesos distribuidos son procesos concurrentes que permiten esta comunicación.

Renaud define cuatro tipos básicos de procesos distribuidos: **filtros, clientes, servidores y pares de terminales**. La diferencia entre ellos, se encuentra ilustrada en la Figura 5.2. Los procesos filtros ejecutan una operación fija en un flujo de datos, pasando los resultados a otro proceso. Los procesos pares son idénticos entre sí para interactuar en una manera cooperativa para trabajar más útilmente. [RENAUD,93].

En la Figura 5.1 se muestran los principales tipos de procesos distribuidos que existen.



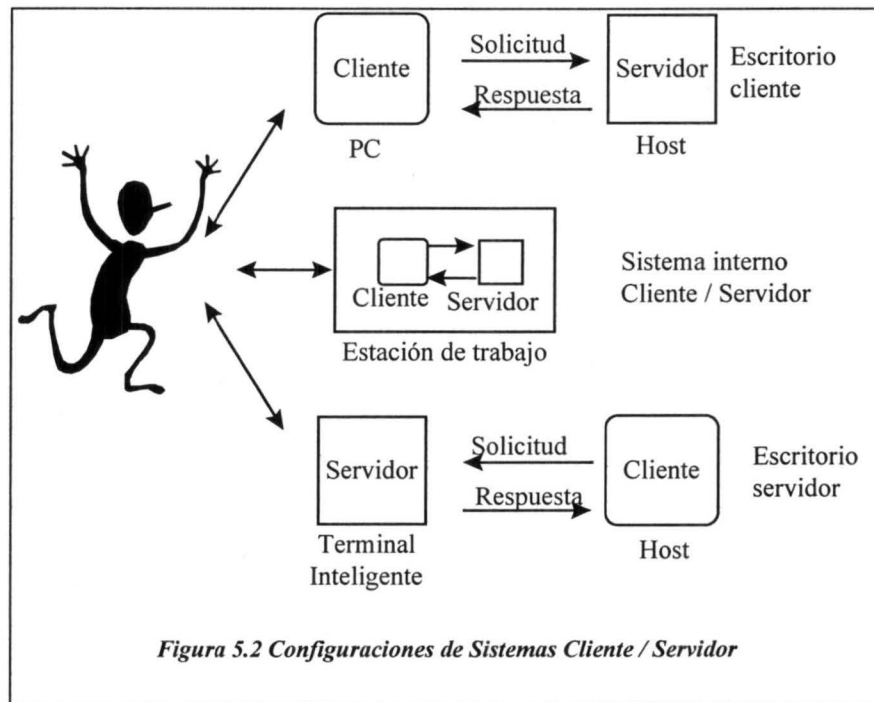
El primer modelo (Peer-to-peer), permite acceder recursos compartidos por otros usuarios y compartir recursos con las estaciones de trabajo locales permitiendo el acceso a otros miembros del grupo. Este esquema denota una comunicación de procesos de muchos a muchos.

En el esquema Peer-to-peer, por ejemplo, los procesos distribuidos ejecutan el mismo programa; en la arquitectura cliente / servidor en cambio, los procesos son independientes. El proceso cliente envía una petición al proceso servidor, el cual responde a esa petición con un resultado. Como su nombre lo indica, los procesos servidores proveen servicios a sus clientes, normalmente de la manera específica de procesamiento que pueden hacer. La interacción entre los procesos cliente y los procesos del servidores son cooperativos, es decir, funcionan por medio del intercambio de transacciones en las cuales el cliente es proactivo y el servidor es reactivo. Esta es la principal diferencia del esquema Cliente / Servidor y los demás paradigmas.

En ambiente cliente / servidor real, es indiferente si los procesos cliente y servidor se ejecutan en la misma máquina o no, sin embargo existen algunos protocolos que no lo soportan.

Desde el punto de vista teórico, no existe ninguna restricción que impida que un procesos servidor pueda ejecutarse en una máquina con menor poder que la que ejecuta un proceso cliente; este tipo de simplicidad es lo que le da el poder a esta tecnología.

La Figura 5.2 muestra tres tipos de configuraciones de sistemas cliente / servidor.



5.1.2 De la teoría, a la práctica

Las implementaciones más comunes de sistemas cliente / servidor se efectúan con procesos cliente y servidor separados por una red local. Sin embargo, esto no limita a que no puedan elaborarse implementaciones en redes amplias o en intranets, incluso.

La siguiente figura ilustra una vista lógica de los sistemas cliente / servidor en la práctica. Es de notarse que el término *cliente* se designa al sistema que ejecuta los *procesos cliente*, y de la misma forma el término *servidor* se emplea para el que ejecuta los *procesos servidor*.

Cabe aclarar que esta vista no limita a que no existan otros sistemas que puedan interactuar con éste; al contrario el poder de esta arquitectura permite la interacción con otros sistemas en línea, por lotes, o incluso con otros sistemas cliente / servidor.

El usuario del sistema interactúa con el cliente, el cual envía peticiones y recibe resultados del servidor. El estilo de interacción no tiene que ser de alguna manera específica.

5.1.3 Características del ambiente cliente / servidor

Atributos del cliente. Normalmente está dedicado a la sesión del usuario; puede interactuar con uno o más servidores para llevar a cabo su trabajo; es proactivo enviando peticiones al servidor.

A un nivel de aplicación el cliente es responsable del mantenimiento y procesamiento del diálogo con el usuario, lo cual implica lo siguiente:

1. Manejo de pantallas
2. Interpretación de comandos y menús
3. Validación de captura de datos
4. Ayuda
5. Recuperación de errores

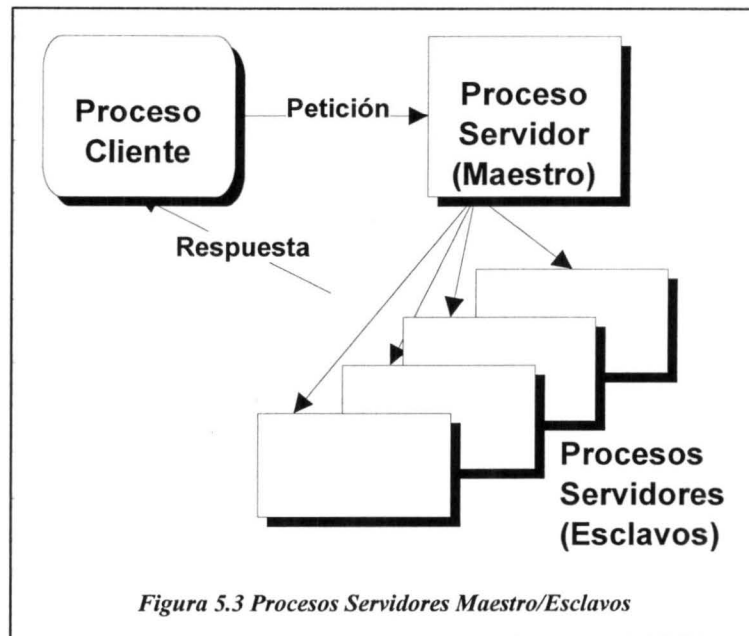
En aplicaciones gráficas el cliente incluye además:

1. Manejo de ventanas
2. Manejo de ratón y entradas de teclado
3. Control de cajas de diálogo
4. Administración de sonido y video

de manera que administrando toda la interacción con el usuario, el cliente oculta todo el diálogo que existe entre él y el servidor a través de la red, dándole al usuario la impresión de que todos los procesos se están ejecutando en su máquina local.

A un nivel de servicios, el cliente es responsable de la detección e intercepción de las peticiones de servicios de la aplicación para redireccionarlas al lugar adecuado.

Atributos del Servidor: Los procesos del servidor son activadas por las peticiones de los clientes. Estos servicios pueden ser provistos por el mismo servidor o indirectamente por procesos dedicados llamados "esclavos", tal y como se muestra en la Figura 5.3.



La función del servidor consiste en ejecutar un conjunto predefinido de transacciones relacionadas entre sí. A menudo existen múltiples servidores proveyendo un repertorio de funciones a la comunidad de clientes en forma simultánea.

Típicamente el procesamiento del servidor incluye todo lo relacionado con el acceso, almacenamiento y organización de datos compartidos, cualquier administración de otros recursos compartidos y la actualización de los datos almacenados. Los recursos compartidos pueden ser datos, CPU, almacenamiento en disco o cinta, impresoras, comunicaciones y administración de memoria.

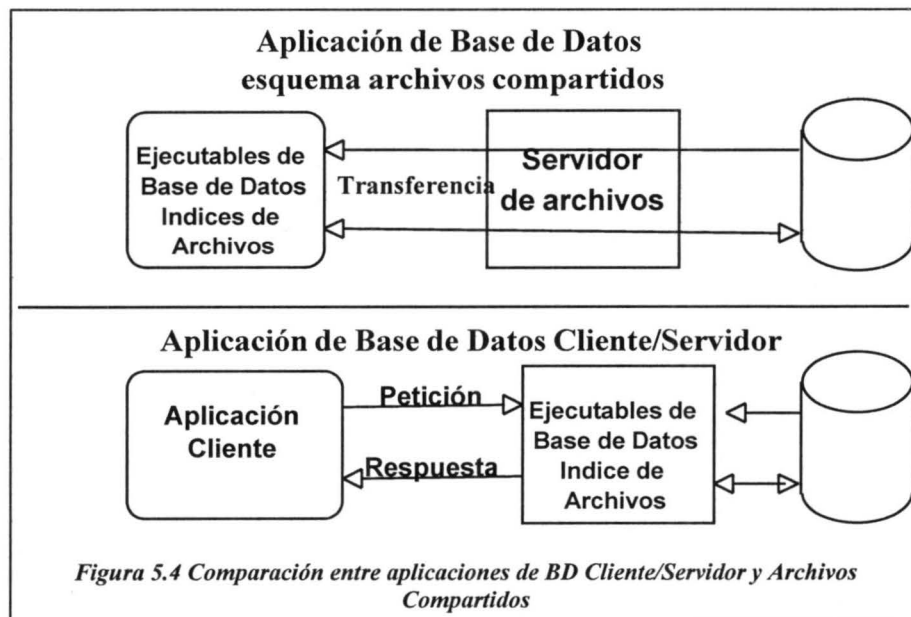
Atributos de la Comunicación: Usualmente existe una relación en la que hay muchos clientes para pocos (o uno) servidores. Es responsabilidad del cliente el localizar e iniciar el intercambio de información con el servidor. La comunicación entre el cliente y el servidor es transaccional y cooperativa, que es muy distinta a la comunicación maestro / esclavo, en la que el proceso maestro gobierna todas las transacciones de los procesos esclavos.

Una característica del tipo de comunicación transaccional es que el servidor regresa *sólo los resultados que son relevantes a la petición del cliente*, lo cual es

favorable en situaciones críticas de tráfico en la red. Esto significa que como la cantidad de datos que es enviada por el servidor (respuesta) es la mínima necesaria para que el cliente complete su tarea, se reduce significativamente la sobrecarga de la red.

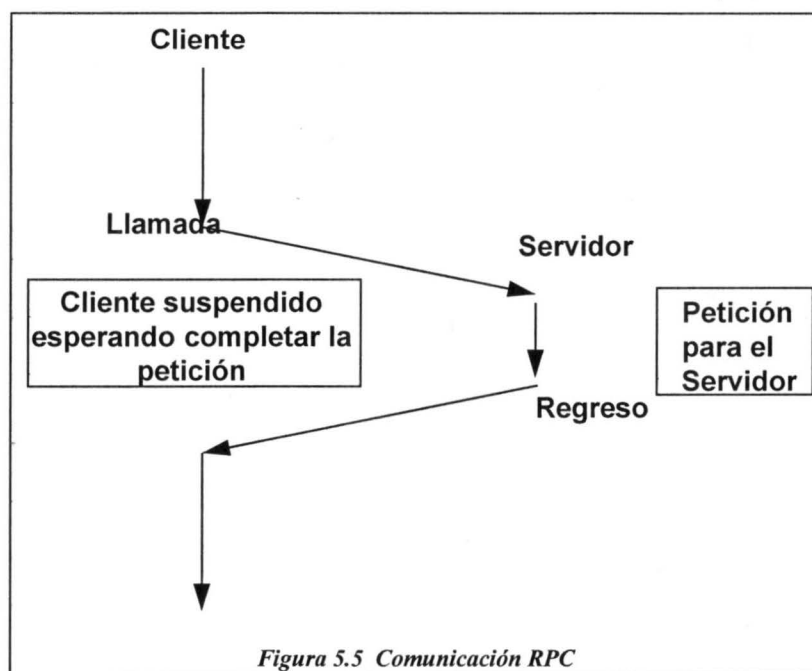
En aplicaciones de red basadas en la tecnología de archivos compartidos, cuando una aplicación va a ejecutarse el servidor de archivos envía el ejecutable entero a través de la red y lo deposita en la memoria de la estación de trabajo. En cambio en la arquitectura cliente / servidor el tráfico de la red consiste sólo en los datos que van a escribirse, pequeños códigos de retorno y algunos encabezados del paquete (véase Figura 5.4).

La naturaleza cooperativa del paradigma cliente / servidor consiste en **que la cantidad de procesamiento que es significativa se realiza en ambos agentes**. A su vez, es necesaria una sincronización, la cual se efectúa a través de mensajes.



La comunicación Cliente-Servidor es implementada usando *llamadas a procedimientos remotos* (RPC, por sus siglas en inglés). En una llamada de RPC, el cliente detiene el procesamiento después de emitir una petición al servidor y esperar su respuesta. Esto es ilustrado en la Figura 5.5; el servidor comienza el procesamiento sólo como una petición del cliente y termina después de cumplir con la petición. Cuando el cliente recibe la respuesta, se

continúa el proceso. Por lo tanto, la sincronización entre el cliente y el servidor está implícita en el mecanismo de traspaso de mensajes.



5.2 Consideraciones para el desarrollo [RENAUD, 93]

La mayoría del procesamiento de las aplicaciones puede ser organizado en tres segmentos: interacción, aplicación y base de datos. El balance del procesamiento en una aplicación es la cantidad de trabajo efectuado por el cliente y por el servidor. Una aplicación está centrada en el servidor si el cliente ejecuta solamente un poco del procesamiento interactivo. En cambio, si el servidor ejecuta algo del procesamiento de base de datos, la aplicación se dice que está centrada en la estación de trabajo. Diseñar una arquitectura cliente / servidor consiste en el balanceo correcto de la carga de procesamiento entre el cliente y el servidor.

La diferencia clave en el desarrollo de un sistema cliente / servidor consiste en que se están desarrollando dos sistemas simultáneamente en lugar de uno, como tradicionalmente se hace. Esto tiene implicaciones tanto en el diseño de la aplicación como en las herramientas que se van a utilizar. El diseño tiene que tomar en cuenta en donde ocurre el procesamiento y los aspectos arquitectónicos del servidor en los referentes a la seguridad de la distribución de los datos.

5.3 MODELO ORIENTADO A OBJETOS

El Modelo Orientado a Objetos (MOO) se basa en el paradigma de programación orientada a objetos. Este paradigma ha tenido gran aceptación en los 90's debido a que es de gran naturalidad buscar objetos en la realidad a modelar.

Esta sección está destinada a introducir al lector en los conceptos básicos relacionados con la Metodología Orientada a Objetos, la cual se discute en la siguiente sección.

Las razones por las cuales es conveniente modelar aplicaciones utilizando la filosofía orientada a objetos son:

- Es más sencillo modelar objetos porque existen en el mundo real. Se piensa en función de objetos
- Permite una interacción sencilla del diseñador con el ambiente computacional, utilizando similitudes que son familiares a la realidad
- Construcción de componentes de software reutilizables y módulos extensibles de bibliotecas de software. Esto permite construir aplicaciones cada vez más complejas y en menor tiempo
- Fácil mantenimiento de las aplicaciones. Por la naturaleza de este modelado las aplicaciones pueden crecer y ser modificadas sin tener que hacer cambios medulares en la estructura del programa

La orientación a objetos pretende satisfacer tanto las necesidades de los usuarios finales como de los desarrolladores de productos de software [KHOSHAFIAN, 90].

5.3.1 Estructura de Objetos.

El modelo orientado a objetos se basa en encapsular código y datos en una única unidad, llamada objeto. La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.

*Definición.- Un **objeto** es una cosa, real o abstracta, acerca de la cual almacenamos datos y los métodos que controlan dichos datos. [MARTIN, 92]* Algunos autores definen un objeto como “una caja negra” la cual recibe y envía mensajes. Esta caja negra contiene código y datos juntos, a diferencia de los lenguajes tradicionales como C.

Un **tipo de objeto** (o clase de objeto) es una categoría más amplia de objeto; de manera que un objeto posee las características de su tipo de objeto más las suyas propias. Esto origina la jerarquía de clases y el concepto de herencia, los cuales se encuentran estrechamente vinculados.

La **jerarquía de clases**¹² se refiere a la forma como los objetos son categorizados en clases y subclases. Un tipo de objeto dado es una clase, su instancia (o sea, su objeto), es una subclase si ésta a su vez tiene instancias más específicas. Por medio de la **herencia** los objetos de los niveles inferiores poseen las características de sus tipos de objetos ascendentes más unas características particulares que los distinguen. De esta manera podemos construir objetos a partir de objetos y cada vez con mayor complejidad.

*Definición.- Los **métodos** especifican la forma en que se controlan los datos de un objeto. [MARTIN, 92]*

El motivo de este enfoque puede ilustrarse considerando una base de datos de documentos¹³ en la que los documentos se preparan usando diferentes plantillas según el propósito para el que fueron diseñadas y según el papel que juegan dentro de la aplicación. Para imprimir un documento debe ejecutarse el formateador correcto en el documento. Bajo un enfoque orientado a objetos, cada documento es un objeto que contiene el texto de un documento y el código que opera sobre el objeto, el formateador en este caso, es el método. Todos los objetos del tipo documento responden al mensaje Imprimir pero lo hacen de forma diferente. Cada documento responde ejecutando el código formateador adecuado. Encapsulando dentro del objeto documento la información acerca de cómo imprimirlo, podemos tener todos los documentos con la misma interfaz externa al usuario.

¹² La jerarquía de clases se explica posteriormente con más detalle.

¹³ Lotus Notes administra bases de datos que almacenan documentos. Ver la sección 3.4.

*Definición.- Se entiende por **encapsulado** al conjunto manejado como unidad conformado por los datos y los métodos de un objeto. El encapsulado se realiza en forma transparente al programador, es decir, no tiene que enterarse de los detalles de la implementación del objeto.*

Por encapsulación un objeto generalmente tiene asociado:

- Un conjunto de atributos que contienen datos acerca del objeto. A su vez, cada valor de un atributo es un objeto.
- Un conjunto de mensajes a los que el objeto responde.
- Un conjunto de métodos, que es un procedimiento o trozo de código para implementar la respuesta a cada mensaje. Un método devuelve un valor en forma de objeto como respuesta al mensaje.

Debido a que la única interfaz de un objeto con otros es el conjunto de mensajes al que responde, es posible modificar la definición de métodos y atributos sin afectar a otros objetos. También es posible sustituir un atributo por un método que calcule un valor.

Ejemplo: Un objeto documento puede contener un atributo de tamaño que contenga el número de bytes de texto en el documento, o bien un método de tamaño que calcule el tamaño del documento leyéndolo y contando el número de bytes.

La capacidad de modificar la definición de un objeto sin afectar al resto del sistema está considerada como una de las mayores ventajas del modelo de programación orientada a objetos.

5.3.2 Jerarquía de Clases.

Normalmente en la realidad a modelar existen muchos objetos similares. Por similar se quiere decir que tienen una naturaleza parecida, por lo que son candidatos a poseer atributos, métodos y responder a mensajes comunes en un contexto orientado a objetos.

Para el caso de los objetos similares, sería un trabajo inútil definir cada uno de ellos por separado. Por tanto, se agrupan para que conformen una clase. A cada uno de estos objetos se le llama instancia de su clase. Todos los objetos de una clase comparten una definición común, aunque difieran en los valores asignados a los atributos.

Ejemplo: En un banco, son objetos los clientes, las cuentas y los préstamos. Una clase incluye:

- Un atributo con valores en un conjunto cuyo valor es el conjunto de todos los objetos que son instancias de la clase.
- Implementación de un método para el mensaje nuevo, el cual crea una nueva instancia de la clase.

Un esquema orientado a los objetos, normalmente requiere un gran número de clases. Sin embargo, a menudo se da el caso que varias clases son similares. Para permitir la representación directa de similitudes entre clases, se utiliza una jerarquía de especialización.

Ejemplo: En un banco, es de esperarse que los Empleados y los Clientes tengan ciertos atributos comunes, como *RFC*, *NOMBRE*, *DIRECCIÓN* Y *TELÉFONO*, sin embargo los Empleados podrían tener un atributo exclusivo, como *SALARIO*, mientras que Clientes tendrán uno como *TASA CRÉDITO*. Aquí los empleados y clientes pueden ser representarse por especializaciones de una clase *Persona*. Los atributos y los métodos específicos para *Empleado* se asocian a la clase *Empleado*, en forma análoga, se actúa para los clientes. Los atributos y métodos comunes para empleados y clientes se asocian a la clase *Persona*.

5.3.3 Herencia.

La Herencia se manifiesta cuando se tiene una clase que es una especialización de otra que se encuentra en un nivel superior en la jerarquía de clases. La clase subordinada (en un nivel inferior) hereda los métodos, mensajes y atributos de la clase superior.

Cuando una clase que hereda de otra necesita incluir un atributo extra, éste se le asocia, y no afecta a la clase superior. Por otro lado, es común que en una

clase que constituye una especialización, se desee modificar alguno de los métodos heredados para adecuar la respuesta a algún mensaje. En este caso, se modifica el comportamiento de esta clase, y el método nuevo se le asocia.

Ejemplo: En el mismo caso del banco, podemos observar que como atributos para la clase persona se tiene *RFC, NOMBRE, DIRECCIÓN Y TELÉFONO*. A su vez los empleados tienen un sueldo, y los clientes una tasa de crédito.

En este caso, un objeto de la clase *Persona* tiene como *ATRIBUTOS RFC, NOMBRE, DIRECCIÓN Y TELÉFONO*, un *Empleado*, hereda todos los anteriores y agrega el atributo *SUELDO*, mientras que el *Cliente* agrega el atributo *TASA CRÉDITO*.

5.3.4 Beneficios de la tecnología orientada a objetos: [MARTIN, 93]

La naturaleza del Modelo Orientado a Objetos tiene inherente un conjunto de características que se traducen en beneficios, algunos de los cuales se presentan a continuación.

- **Reutilización.** Los tipos de objetos están diseñados para que se reutilicen en muchos sistemas. El objetivo de la TOO es lograr la reutilización masiva de objetos a partir de bibliotecas cada vez más grandes.
- **Estabilidad.** La reutilización de objetos supone bibliotecas de objetos que ya han sido probados en cada utilización, y que por lo tanto brinda una mayor seguridad de correcto funcionamiento al seguir siendo implementados.
- El diseñador piensa en términos del **comportamiento de objetos** y no en detalles de bajo nivel. Gracias al *encapsulado*, las clases complejas son fáciles de utilizar. Basta con entender el comportamiento del objeto (sin conocer los detalles de la implementación) para que éste pueda ser utilizado.
- Se construyen **clases** cada vez **más complejas**. Esto debido a la habilidad de construir clases a partir de clases; de esta manera y por las bibliotecas de objetos, los desarrollos ya no tendrán que partir de cero.

- **Confiabilidad.** Es probable que el software construido a partir de clases existentes ya probadas tenga menos fallas que el software elaborado a partir de cero.
- **Nuevos mercados para el software.** Las compañías de software pueden proporcionar bibliotecas de clases para áreas específicas y que se adapten a las necesidades del cliente.
- **Un diseño más rápido y de mayor calidad.** Esto se debe a que las aplicaciones se crean a partir de componentes ya existentes, los cuales ya fueron probados en sistemas anteriores. Muchos de ellos están contruidos de tal manera que se pueden adaptar a un diseño particular, lo cual agiliza el proceso del diseño.
- **Integridad.** Las estructuras de los datos sólo se pueden utilizar con los métodos específicos del objeto, de manera que ningún evento o agente externo pueda modificarlos y provocar inconsistencias o alteraciones de los datos no advertidas.
- **Programación más sencilla.** Los programas se conjuntan a partir de piezas pequeñas, algunas preexistentes.
- **Mantenimiento más sencillo, mejores interfaces, facilidad para migración a otras plataformas.**

5.4 ANALISIS Y DISEÑO ORIENTADO A OBJETOS

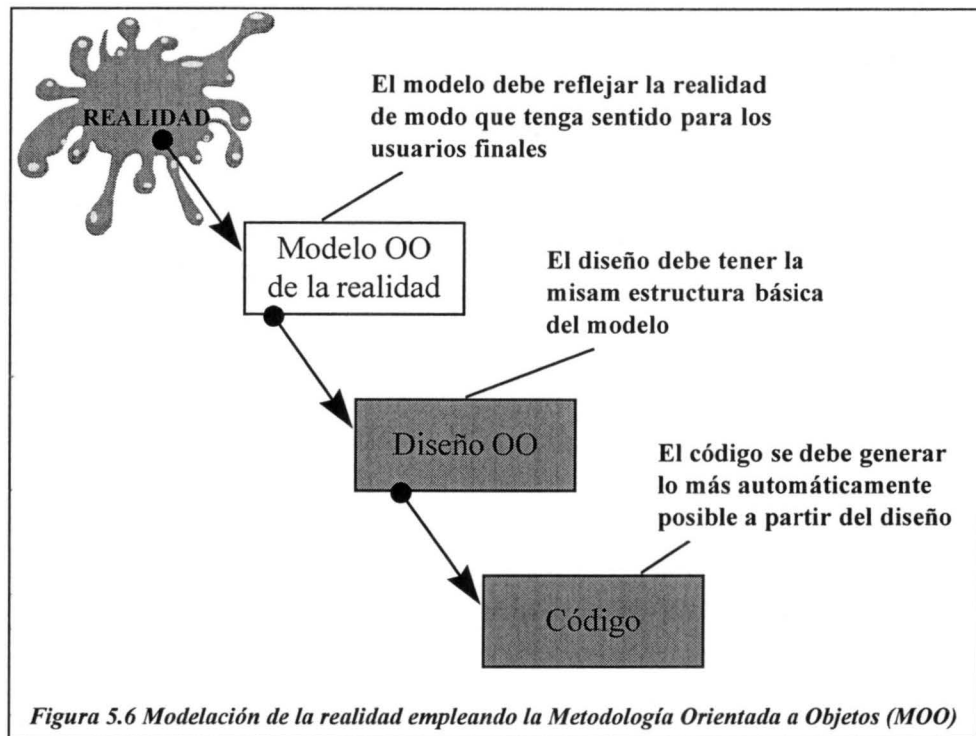
Tradicionalmente, un modelo es una representación de la realidad. Este modelo nos sirve para comprenderla mejor y analizar cómo es afectada por el entorno. La presente sección apoya los objetivos personales No. 4 y 5, y tiene como finalidad que el lector comprenda los elementos básicos de esta metodología a fin de que los diagramas generados durante el análisis y el diseño del proyecto le sean comprensibles.

Como ya se dijo, lo más natural es pensar en objetos, es por eso que en el análisis orientado a objetos modelamos el mundo en término de objetos y lo que les sucede. Lo mismo sucede con el diseño y la programación. De esta manera, si la realidad que estamos modelando cambia, será mucho más fácil para nosotros implementar esos cambios.

La mejor solución que podemos brindar a nuestros usuarios podría consistir en capturar el punto de vista del usuario con respecto al mundo y traducirlo en software de la manera más automática posible. Obsérvese la Figura 5.6

Con las metodologías tradicionales de desarrollo los cambios introducidos por el inestable ambiente de las empresas son difíciles de implementar. Con el análisis y diseño OO ¹⁴ se pretende expresar en el modelos las reglas de las empresas de la manera más explícita posible para traducirlas en diseño y código; así si las reglas cambian, el modelo puede ser fácilmente cambiado.

¹⁴ OO: Orientado a Objetos.



5.4.1 Dos Tipos De Modelos [Martin, 93]

Existen dos tipos de modelos representados a través de esquemas que se utilizan para el análisis orientado a objetos:

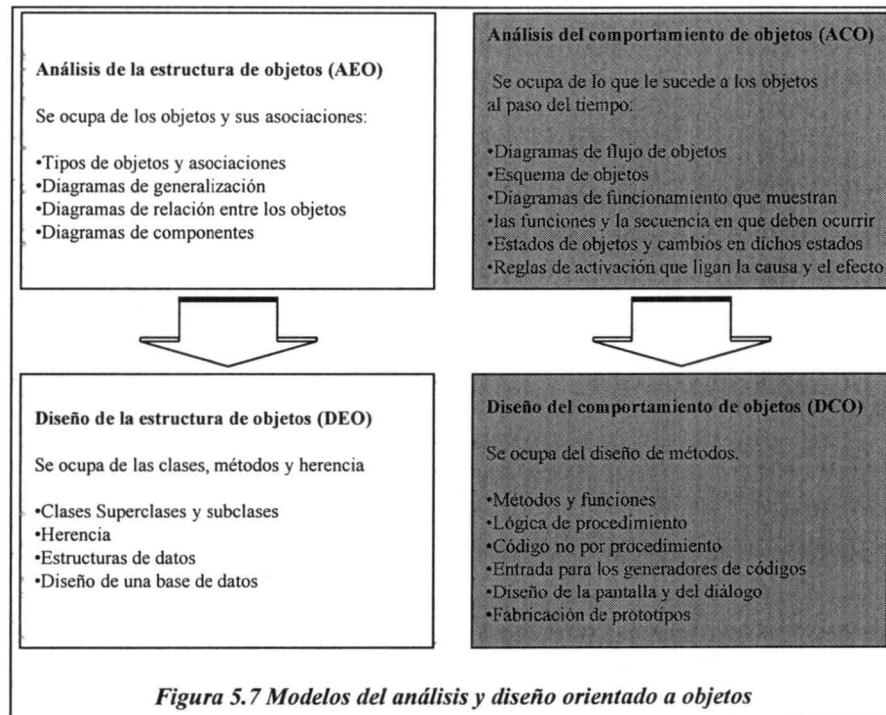
1. Un modelo referente a los tipos de objetos y su estructura
2. Un modelo referente a lo que le ocurre a esos objetos

Tanto el análisis como el diseño orientado a objetos contemplan ambos modelos a través del análisis y diseño de la estructura y comportamiento de objetos.

El análisis de la estructura de objetos (AEO), y el diseño de la estructura de objetos (DEO) se refiere a la definición de los tipos de objetos, clases, relaciones entre los objetos y la herencia.

El análisis y diseño del comportamiento de objetos (ACO y DCO, respectivamente) se ocupa del comportamiento de los objetos y su transformación en el tiempo.

De esta forma tenemos proceso consistente de desarrollo de un producto, tanto analistas como diseñadores y hasta los programadores y los mismos usuarios piensan en función de objetos, cosa que no sucede en los desarrollos tradicionales donde los modelos conceptuales de las etapas de análisis y diseño difieren (Figura 5.7).



5.4.2 Análisis de la Estructura de Objetos [Martin, 93]

El análisis de la estructura de objetos define las categorías de los objetos que percibimos y las formas en que los asociamos. Los modelos que se crean durante esta etapa son una guía para la definición de clases y sus estructuras de datos en el diseño de la estructura y comportamiento del objeto.

El AEO se ocupa de identificar tres tipos básicos de información:

1. **¿Qué son los tipos de objetos y cómo se asocian?** La identificación de los objetos y sus asociaciones se representan mediante esquemas de objetos. Esta información guía al diseñador en la definición de clases y estructuras de datos.

Es muy importante identificar los tipos de objetos que clasifican los objetos de nuestro entorno. En adición, también debemos modelar la forma como los objetos serán asociados.

En el análisis, es útil utilizar alguna nomenclatura para identificar la cardinalidad de la asociación entre los objetos.

2. **¿Cómo se organizan los tipos de objetos en supertipos y subtipos?** Las jerarquías de generalización se pueden organizar en diagramas e indicar al diseñador las direcciones de herencia.

Una de las maneras como el hombre organiza su conocimiento en forma natural es en jerarquías. De lo general, a lo específico.

Las jerarquías de generalización son particularmente importantes en el desarrollo OO porque describen el mundo del sistema como lo es en el mundo real, además que indica las direcciones de la herencia entre las clases.

3. **¿Cuál es la composición de los objetos complejos?** Se pueden elaborar diagramas de jerarquías compuestas. La composición guía al diseñador en la definición de mecanismos que controlen adecuadamente a los objetos dentro de otros objetos.

5.4.3 Análisis del Comportamiento de objetos

Los esquemas que se realizan en esta fase muestran los eventos¹⁵, su secuencia y la forma como cambian el estado de los objetos. El AEO y el ACO están ligados muy estrechamente, se desarrollan en conjunto para formar modelos y diseños integrados.

¹⁵ Un evento es el cambio de estado de un objeto.

La información que se identifica en el ACO es la siguiente:

1. **Estados en los que puede estar un objeto.** Un objeto puede tener varios estados, los cuales se conocen como estados posibles dentro del ciclo vital del objeto.

El estado de un objeto es la colección de los tipos de objeto que se aplican a él más la colección de asociaciones que tiene un objeto. (Para mayor información sobre este punto consúltese la bibliografía, [MARTIN, 93]).

2. **Transiciones de estado que se pueden dar.** Son determinadas en el diagrama de transición de estado.
3. **Eventos que ocurren.** Se traza un esquema de eventos para mostrar la secuencia de eventos.
4. **Operaciones que se llevan a cabo.** Se traza un esquema de eventos para mostrar la secuencia de operaciones. El término operación se refiere a una unidad de procesamiento que puede ser solicitada. Esta unidad se implanta a través de un *método*. El método es la especificación de cómo llevar a cabo la operación.
5. **Interacciones que ocurren entre los objetos.** Se elabora un diagrama para mostrar los mensajes que se transfieren entre las clases.
6. **Cuáles son las reglas de activación.** Qué operaciones son activadas a raíz de un evento. Las reglas de activación definen la relación entre la causa y el efecto, es decir, siempre que ocurra un evento de cierto tipo, las reglas de activación invoca a una operación ya definida.
7. **Cómo se representan las operaciones en los métodos.** Se utilizan los diagramas, enunciados declarativos u otros medios para determinar los métodos con el suficiente detalle para generar el código.

5.4.4 Diseño de la estructura y el comportamiento de un objeto (DEO y DCO)

Al igual que en el análisis, el DEO y el DCO se encuentran íntimamente entrelazado. La razón de esto radica en que los lenguajes de programación orientados a objetos (LPOO) tienen estructuras de datos y métodos combinados dentro de las clases.

Los componentes del DEO y el DCO son los siguientes:

1. **Clases que se propone implantar.** Para esta decisión la guía son los tipos de objetos en el AEO.
2. **Estructura de datos de cada clase.** Para representarlas se emplean diagramas de estructura de datos.
3. **Operaciones y métodos que están disponibles para de clase.** Se especifican ambos.
4. **Forma de implementación de la herencia.** Se identifican cuáles datos y métodos se van a heredar para qué clases.

5.5 LOTUS NOTES

Notes, como es llamado comúnmente, es un sistema administrador de aplicaciones destinado a encontrar y compartir información para hacer el trabajo de una manera más sencilla y trabajar mejor con otras personas. Si se cuenta con una infraestructura de red, estas personas podrán estar alejadas geográficamente y trabajar "juntas" tan bien como si lo estuvieran físicamente.

Esta sección tiene el objetivo de que el lector conozca la plataforma de desarrollo y se familiarice con los términos relacionados a ella. Su contenido apoya principalmente los Objetivos Personales No 6 y 8 y la mayoría de los Objetivos del Sistema.

Notes guarda información en bases de datos. Cada información tiene un tema específico, y el creador decide qué información contendrá la base de datos en función del tema elegido. La base de datos está organizada por documentos, cada uno contiene información específica en base a una plantilla o forma.

Lotus Notes permite hacer el desarrollo de sistemas estratégicos, administra la información de un grupo de trabajo; de tal manera que junta, organiza, comparte y procesa la información que generan los miembros de un grupo. Por sus características, Notes es favorable para trabajar con la arquitectura **Cliente-Servidor**.

5.5.1 Aplicaciones de Lotus Notes

Una aplicación de Notes es una implementación de datos y programas que soportan una función o propósito del negocio. Estas aplicaciones nos permiten crear, comunicar, compartir y seguir la pista de *información orientada a documentos* a través de redes. En el ambiente de Notes, los usuarios accesan la información a través de aplicaciones consistentes en una o más bases de datos.

Los usos de las aplicaciones de Notes pueden ser muchos:

1. Enviar y recibir mensajes en correo electrónico
2. Leer sobre las últimas noticias en el mundo de la industria

3. Encontrar inventario y automáticamente crear nuevas órdenes cuando las que tenga no sean suficientes
4. Planear proyectos complicados
5. Colaborar en proyectos con otras personas en otros sitios.
6. Hacer formularios para aquellos que necesitan verlos y aprobarlos.
7. Compartir y discutir información importante para el trabajo (disminuyendo la cantidad de reuniones físicas)
8. Acceso a Notes desde lugares remotos
9. Acceso a Internet

5.5.2 Tipos de aplicaciones de Notes

Normalmente se clasifican en una o más de los siguientes tipos:

1. Broadcast
2. Referencia
3. Tracking o seguimiento
4. Discusiones
5. *Workflow* (Flujo de trabajo)

Unos ejemplos podrían ser aplicaciones destinadas a la Administración de Proyectos, Discusiones sobre la aprobación de nuevos productos,

5.5.3 Aplicaciones de *Workflow* o “Flujo de Trabajo”

Definición

Ultimus [ULTIMUS, 96] define *Workflow* (o flujo de trabajo) como “cualquier tarea que puede desempeñarse en serie o en paralelo por dos o más miembros de un equipo de trabajo para alcanzar una meta común”, donde por tarea se entiende a una actividad relacionada con la misión del negocio; “en serie o en paralelo” indica la forma como se realizan estas actividades”, y, “dos o más miembros” se requieren para coordinarse e interactuar para llevar a cabo las actividades del negocio.

Otra definición podría ser “**Workflow**. Las actividades e interacciones requeridas para llevar a cabo un proceso de negocio.” [LOTUS, 97]

Los términos que se intervienen en esta definición son:

- **Proceso o reglas del negocio.** Los cuales son una secuencia de pasos que se requieren para alcanzar una meta específica, como el proveer soporte al cliente por ejemplo.
- **Actividad.** Un paso de la secuencia de pasos en un proceso de negocio, como el llenar una forma. Las actividades constituyen el “trabajo” en el flujo de trabajo.
- **Interacción.** Se refiere a un paso o secuencia de pasos que traslada información de una persona, lugar o condición a otro. Las interacciones constituyen el “flujo” en el flujo de trabajo.

Automatizando el *Workflow*

Para automatizar el *Workflow*, muchas de las actividades y sus interacciones dentro del proceso del negocio que originalmente se efectuaban utilizando papel son reemplazadas por el medio electrónico para guardar el trabajo de los usuarios.

Algunos procesos basados en papel que son candidatos por sí mismos para ser automatizados son;

1. Envío de documentos de un usuario a otro
2. Asignación de tareas y recursos a un proyecto
3. Compartir información
4. Envío de recordatorios en intervalos predefinidos

Cada aplicación que a menudo involucre las actividades de una cantidad grande de usuarios, algunas veces de diferentes grupos de trabajo y usualmente a través de bases de datos múltiples.

Tipos de aplicaciones de *Workflow*

Este tipo de aplicaciones cubren un gran rango. Una aplicación de *Workflow* puede incluir uno o más de otro tipo de aplicaciones, como las de discusión, referencia o mapeo.

5.5.4 Implementando el Workflow en Notes

En Notes el término **Workflow** se refiere al movimiento de documentos e información a través de una aplicación.

Aplicaciones como seguimiento a información y sistemas de aprobaciones basadas en papel usualmente son descritas como sistemas de *Workflow*.

Envío y compartición de información

El modelo de Notes para *Workflow* puede manipular tanto “procesos a la medida¹⁶” como procesos continuos.

- El **modelo de envío** está basado en el ruteo de formas: información que es enviada a los usuarios. Este es un ejemplo de procesos a la medida.
- El **modelo de compartición** está basado en bases de datos compartidas: usuarios que utilizan datos en forma simultánea.
- Los **modelos híbridos** son una combinación de los modelos; esta es la mejor aproximación para muchas aplicaciones, que comparten bases de datos, pero incluyen funciones de correo para alertar a los usuarios de nuevas actividades.

Resumen de las características del *Workflow* en Notes

Notes provee una variedad de características que permiten definir y automatizar el *Workflow* en aplicaciones. Existen herramientas de automatización como acciones y agentes, las cuales facilitan el ruteo de documentos y mensajes para automatizar los procesos del negocio.

¹⁶ Se entiende por “procesos a la medida” como aquellos que fueron desarrollados para resolver una problemática específica del negocio; se les conoce también como soluciones ad hoc.

5.5.5 Base de datos de Notes

Una base de datos de Notes almacena los documentos y elementos del diseño para esa base de datos. Los elementos del diseño incluyen vistas, formas y campos. Los documentos son registros individuales de información.

Formas

La forma es elemento fundamental de diseño en una base de datos de Notes. Una forma es una plantilla de documento que provee una estructura para la captura de los datos. Cada vez que los datos son capturados en una forma, la información es almacenada como un documento en la base de datos.

Campos

Un campo es una porción del área de una forma que contiene una unidad simple de información. Cada campo en una forma determina el tipo de información que el campo puede contener.

Existen 4 tipos básicos de campos en Notes:

- Texto
- Texto enriquecido (Rich Text)
- Número
- Tiempo o fecha

El campo texto está compuesto por letras, puntuación, espacios y números que no pueden ser utilizados matemáticamente. Los campos de tipo texto enriquecido permiten texto y gráficas; esto incluye también botones, anexos (attachments) y objetos embebidos. El campo número contiene información que puede ser usada matemáticamente y el campo tiempo contiene información de fechas y horas.

Existen algunos tipos adicionales de datos que facilitan el desarrollo de aplicaciones y tienen más bien una definición funcional. Los nombres se mencionarán en inglés dado que es su identificación nativa.

- **Keywords.** Permiten crear una lista, dejando al usuario seleccionar una opción de algunas alternativas predefinidas. Son de tres tipos: 1) Lista de diálogo, 2) Radio button y 3) Checkbox
- **Names.** Se utilizan para especificar quien puede leer y editar los documentos creados a partir de la forma que los contiene.

Documentos

Un documento es una colección de datos que fueron creados utilizando una forma. Los datos también pueden ser introducidos en un documento a través de programación utilizando fórmulas.

Vistas

Una vista es como una tabla de contenido dinámica. Permite a los usuarios acceder los documentos de Notes en una base de datos.

Las vistas están compuestas de columnas y renglones que presentan a los usuarios una lista de los documentos mostrando los campos asociados con cada uno.

En síntesis, la estructura de un documento de Notes está definido por una forma, la cual muestra un conjunto de información a través de campos. Los documentos pueden ser accedidos a través de vistas, las cuales están dispuestas en forma de árbol y muestran información clave de los documentos por medio de columnas.

Agentes

Los agentes son fórmulas predefinidas que actúan en un conjunto predeterminado de documentos. Trabajan en una base de datos para administrar documentos, enviar mensajes y cambiar valores de los campos. Son almacenados dentro de la base de datos para la cual fueron creados.

Los agentes pueden iniciar su ejecución tanto por predeterminado tiempo como evento, o incluso manualmente desde un menú.

Algunos detonadores de los agentes son:

- Por tiempo:
 - Cada hora, día, mes o año
 - En fechas y horas específicas
- Por evento:
 - Cuando llega un nuevo mensaje de correo
 - Cuando documentos son creados o modificados
 - Cuando documentos son incrustados

Acciones

Para automatizar el *Workflow* se pueden crear acciones sobre las formas y sobre las vistas. Estas acciones, llamadas según el caso acciones de forma y acciones de vista, pueden utilizar acciones simples de Notes para enviar documentos a otras bases de datos o usuarios o puede requerir fórmulas más complejas especificando condiciones y múltiples destinos.

Lotus Script

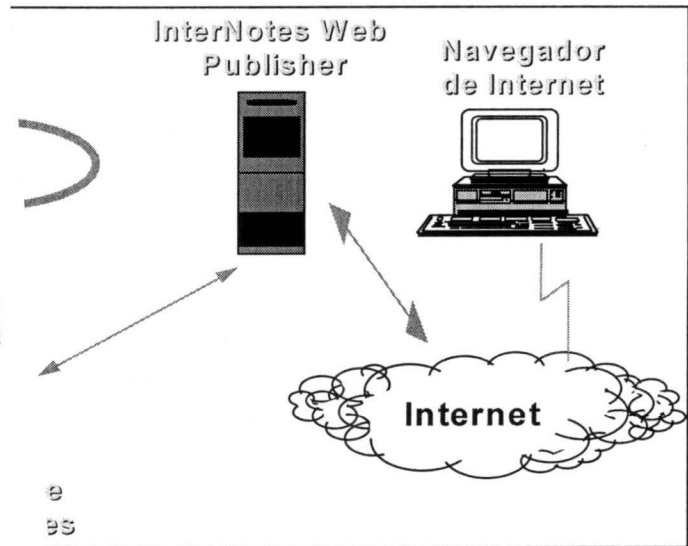
Lotus Script es una herramienta poderosa de desarrollo en forma de lenguaje. Este lenguaje (similar a Visual Basic) maneja un ambiente de programación orientada a eventos. Este tipo de programación es favorable para el empleo de una metodología orientada a objetos.

El lenguaje nativo de desarrollo de Notes es el *Notes Formula Language*, el cual provee al desarrollador de fórmulas, variables y operadores que permiten la programación de eventos a través de un conjunto de funciones predefinidas.

5.5.6 Notes en Internet [CHAVEZ, 96]

Lotus Corp. desde que liberó su versión 3.0 ha ido integrando el soporte directo de TCP/IP en su sistema de Lotus Notes en 1993. El siguiente año liberó un gateway para establecer comunicación con los protocolos SMTP/MIME. En 1995 liberó la primera versión beta del Internotes Web Publisher. En la actualidad Lotus Notes brinda soporte para los protocolos estándar de Internet a través de su versión 2.0 de Domino.

er, el cual es una respuesta temprana a los tos de Internet, se encarga de convertir s, vistas y bases de datos de Lotus Notes en ósito de desarrollar aplicaciones que también Internet, con una interfaz muy similar a la original Sin embargo esta conversión se traduce en una ra para los clientes de Notes, y otra para los que ura 5.8 muestra la arquitectura de funcionamiento



Funcionamiento de InterNotes Web Publisher

...e permite que el servidor de Lotus Notes se ¹⁷, el cual permite que cualquier cliente pueda es de forma segura y transparente. Integra y los protocolos estándar de Internet bajo las plataformas de Notes.

[LOTUS, 96c]

Con Domino, las aplicaciones desarrolladas para la plataforma Lotus Notes, pueden mostrar a los clientes de Web vistas, formas, documentos y campos en forma muy similar a como es mostrada en su plataforma original. Domino no requiere de un espacio adicional para las aplicaciones que se publican en Internet; utiliza las mismas aplicaciones almacenadas en su formato nativo y las

¹⁷ Web es la forma común de llamar a la red internacional Internet.

CONSULTA LAS POLÍTICAS DE BIBLIOTECA EN: www.udem.mx/biblioteca
 TEL: 8124 1389
 FIRMA
 \$5.00 por día u hora, por cada ítem.
 Después del vencimiento acepto pagar.
 Acepto el plazo de préstamo.
 GUARDE LA PAPELETA PARA ACLARACION
 TOTAL DE ÍTEMS: 1
 FECHA DE VENCIMIENTO: 08/10/02
 COLECCION: ESCA
 CLASIF. 040.6588 E43p. 1897
 ÍTEM: 33403001500507
 TÍTULO: Planeación de campaña publicitaria
 USUARIO: 000058458
 NOMBRE: Alvarado Garza, Ana
 ESTACION: reserva2
 24/09/02 11:52AM
 Biblioteca Central
 COMPRABANTE DE PRESTAMO
 UNA INSTITUCION ACREDITADA POR SACS
 UNIVERSIDAD DE MONTERREY

transforma en forma dinámica. Por si fuera poco, el servidor de Domino, también puede ser utilizado como servidor de Internet convencional, permitiendo así el desarrollo de Web sites con la construcción de páginas en lenguaje HTML ¹⁸.

Domino integra además el protocolo de seguridad llamado *Secure Socket Layer (SSL)*, el cual permite establecer comunicaciones privadas seguras con los siguientes beneficios:

- **Privacidad.** La información HTTP ¹⁹ es encriptada desde la fuente hasta el destino.
- **Validación de mensajes.** La información transmitida es acompañada de un mensaje codificado, de forma que se pueda detectar cualquier intento de plagio de la información.
- **Autenticación de servidores.** Los servidores bajo este protocolo cuentan con una firma digital, que acompaña a los mensajes, de manera que el cliente pueda asegurarse de que la identidad del servidor es correcta.

La Figura 5.9 muestra el ambiente de trabajo con Domino.

¹⁸ HTML es un lenguaje nativo para aplicaciones desarrolladas para Internet. Su acrónimo se lee como Hyper Text Marked Language.

¹⁹ HTTP es el protocolo de comunicación de Internet. Su acrónimo se lee como Hyper Text Transfer Protocol.

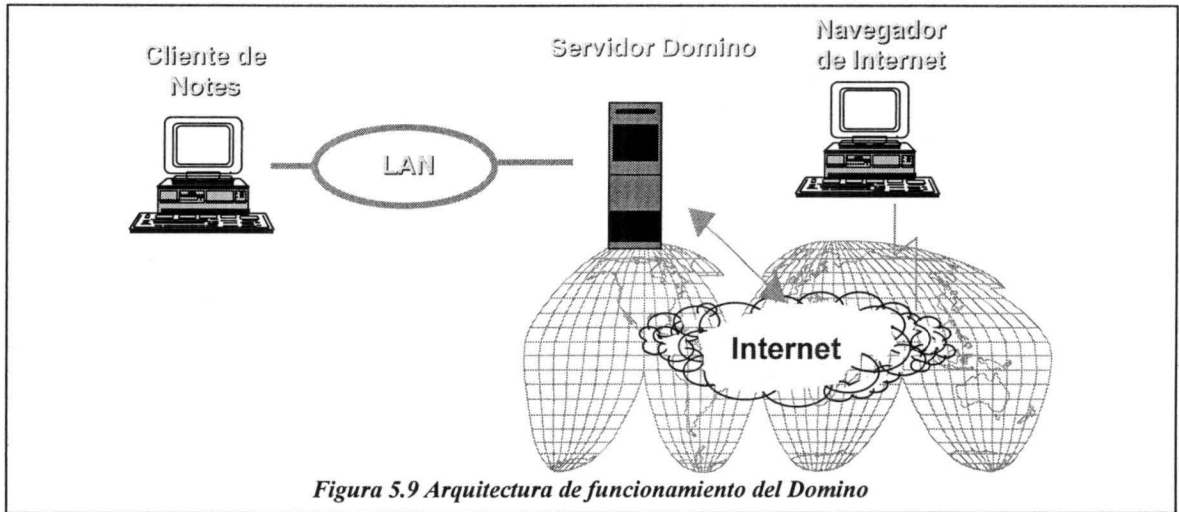


Figura 5.9 Arquitectura de funcionamiento del Domino

5.6 TECNOLOGÍA ODBC

En el ámbito de la Tecnología de Información (TI) cada proveedor provee sus propios estándares para la interfaz de sus productos con el exterior. Existen organismos cuya misión es el establecimiento de estándares internacionales para la comunicación de datos, por citar algunos: ANSI e ISO. Múltiples productos ofrecen aproximaciones con estándares, sin embargo la naturaleza tan distinta de las diferentes plataformas que existen en el mercado no ha permitido que exista un sólo estándar único y absoluto para la comunicación de datos entre plataformas.

Esta sección describe algunos conceptos básicos de la Tecnología de Conectividad de Bases de Datos Abiertas conocida como ODBC y tiene como propósito revelar al lector los alcances y limitaciones de dicha tecnología en materia de comunicaciones conforme los Objetivos del Sistema No 6 y 7.

Un esfuerzo de Microsoft en el establecimiento de estándares en la conectividad a bases de datos distintas consiste en la tecnología llamada ODBC (Open DataBase Connectivity) integrada a través de Windows en lo que se llama ODBC API²⁰. Un API generalmente es un método por el cual un programa puede obtener acceso a modificar el sistema operativo [JENNINGS, 94]. En Windows, los API's son alrededor de 750 funciones provistas por DLL's de Windows que proveen la habilidad a las aplicaciones para manipular elementos del sistema como leer el teclado o interpretar los movimientos del ratón.

El ODBC API consiste en un grupo de archivos DLL de Windows que incluyen conjuntos de funciones que proveen los servicios fundamentales de bases de datos para todos los tipos de bases de datos cuyos manejadores (drivers) están disponibles.

²⁰ API significa Application Programming Interface

5.7 Servicios de ODBC API

Existen dos servicios fundamentales que provee el ODBC API:

1. Instalación, configuración y eliminación de fuentes de datos ODBC. Por una fuente de datos se conoce como una conexión a una base de datos. Se requiere un manejador o *driver* para cada tipo diferente de base de datos para la cual se desea establecer una conexión desde una aplicación. El ODBC API está diseñado principalmente para utilizarse con administradores de bases de datos relacionales (RDBMS) en cliente / servidor, sin embargo también es posible establecer una conexión hacia archivos de bases de datos locales, hojas de cálculo y archivos planos. ODBC utiliza la biblioteca ODBCINST.DLL para configurar y remover las fuentes de datos.
2. Administración de la comunicación de consultas (*queries*) y otros estatutos de SQL del cliente (*front end*) hacia el servidor (*back end*), y transferencia de los resultados de la consulta o confirmaciones de ejecución de las mismas en el sentido contrario (del servidor al cliente). El *ODBC driver manager*, ODBC.DLL, abre el manejador ODBC para la fuente de datos y envía los estatutos de SQL al manejador. Una vez que el RDBMS ha procesado la consulta, el manejador ODBC devuelve la tabla resultado a través del ODBC.DLL hacia la aplicación que lo solicitó.

El ODBC API implementa SQL como una interfaz a nivel llamada (CLI, call-level interface). Un CLI emplea un conjunto de funciones estándar para ejecutar tareas específicas, como la traducción de consultas SQL del estándar SQL (ANSI SQL) al dialecto usado por el RDBMS, la representación de los tipos de datos de los campos del RDBMS por un conjunto extendido de tipos de datos de los campos de SQL-92²¹, y el manejo de las condiciones de error.

²¹ SQL-92 es una convención formal sobre la gramática de SQL, para mayor información consúltese [JENNINGS, 94]

5.8 Tipos de manejadores de ODBC

Los manejadores de ODBC se clasifican en alguna de las siguientes categorías:

1. *Single-tier*, los cuales están diseñados para usarse con administradores de bases de datos (DBM) que no tienen la capacidad de procesar estatutos del ANSI SQL. Este tipo de manejadores traducen los estatutos ANSI SQL a una larga serie de instrucciones de bajo nivel que operan directamente en los archivos constituidos en la base de datos. Un ejemplo de este manejador es el de Microsoft Q+E, el cual establece la conexión hacia bases de datos como dBASE, FoxPro, Paradox, Btrieve, Excell y archivos tipo texto.
2. El *Multiple-tier* cuenta con el RDBMS de cliente-servidor para procesar los estatutos de SQL. Utilizar el servidor (back-end) para procesar las consultas es mucho más eficiente que los manejadores de la categoría *single-tier* en el caso de un ambiente multiusuario. El tráfico de la red es minimizado debido a que el servidor únicamente devuelve los renglones especificados por la consulta.

5.9 Niveles de conformación del ODBC API

Existen 3 niveles definidos que conforman el ODBC API: nivel núcleo (*core level*), nivel 1 y nivel 2.

El *Core level* provee la conexión hacia la base de datos, prepara y ejecuta los estatutos de SQL, recibe los resultados de la consulta, efectúa las transacciones y manipula la información de error.

El nivel 1 es muy similar, agrega la conectividad a fuentes de datos con diálogos diseñados específicamente para el manejador de ODBC. A través de estos diálogos es que se establecen las opciones de configuración del manejador.

El nivel 2 añade la habilidad de listar y examinar las conexiones a fuentes de datos disponibles para el cliente, recupera los resultados de la búsqueda en un

formato de arreglo, y obtiene información adicional del catálogo como privilegios, llaves y listas de procedimientos almacenados (*stored procedures*).

5.10 Niveles de Conformación del ODBC SQL

El ODBC API especifica tres niveles de conformación para la gramática de SQL: mínimo, núcleo y extendido. Los niveles de conformación de SQL definen las palabras reservadas del ANSI SQL que pueden ser procesadas por los manejadores de ODBC. La Tabla 5.2 lista las palabras reservadas del Lenguaje de Manipulación de Datos (DML), el Lenguaje de Definición de Datos (DDL) y el Lenguaje de Control de Datos (DCL) requeridas para cada nivel y sus expresiones. Los niveles de gramática mayores incluyen la gramática de los niveles anteriores.

Nivel	DML	DDL/DCL	Tipos de datos	Expresiones
Mínimo	SELECT, INSERT, UPDATED, SEARCHED, DELETE SEARCHED COMMIT, TRANSACTION, ROLLBACK TRANSACTION	CREATE TABLE, DROP TABLE	CHAR	Aritmética Simple
Núcleo	Full SELECT, positioned UPDATE, positioned DELETE	ALTER TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, DROP VIEW, GRANT, REVOKE	VARCHAR, DECIMAL, NUMERIC, SMALLINT, INTEGER, REAL, FLOAT, DOUBLE	Subqueries y funciones de agregación.
Extendido	LEFT OUTER JOIN, RIGHT OUTER JOIN	Estatutos SQL por lotes, y procedimientos almacenados	LONG, VARCHAR, BIT, TINYINT, BIGINT, BINAR Y, VARBINARY, LONG, VARBINARY, DATE, TIME, TIMESTAMP	SUBSTRING, ABS

Tabla 5.2 Gramática de SQL, tipos de datos y expresiones soportadas por los niveles de conformación de ODBC

6. ANÁLISIS DEL SISTEMA

El Análisis Orientado a Objetos no es un enfoque que modela la realidad sino un modelo de la forma en que las personas comprenden y procesan la realidad [MARTIN, 94].

El presente capítulo presenta en Análisis del Sistema según la metodología que propone James Martin apoyando lo expuesto en los Objetivos Personales No 4 y 5. En primer término se muestra el Análisis de la Estructura de Objetos (AEO) y en seguida el Análisis del Comportamiento de Objetos (ACO). Se recomienda estudiar ambas partes en ese orden, ya que los resultados de la primera son base para la última.

6.1 Análisis de la Estructura de Objetos (AEO)

El análisis de la estructura de los objetos define las categorías de objetos percibidas y la forma en que se asocian.

Esta sección presenta los modelos que se crean durante el AEO en base a la metodología de James Martin. Estos modelos son los siguientes:

1. Identificación de tipos de objetos.
2. Esquema de asociaciones entre objetos.
3. Jerarquía de generalización de los tipos de objetos.

6.1.1 Identificación de Tipos de Objetos

Se identificaron 6 tipos de objetos; tres de ellos externos al sistema y los demás son internos.

I. Tipos de Objetos Externos:

- A. **Usuario.** Es la persona que interactúa directamente con el sistema. La instancia de este tipo de objeto puede interactuar desde un cliente LN o desde un browser conectado a la intranet.
- B. **Tablas AS/400.** Este es un tipo de objeto compuesto de 5 tablas, las cuales contienen la información arrojada por el proceso de Impresión de Recibos de la Nómina JDE.
- C. **Archivos Novell.** Este tipo de objeto también es compuesto. Consta de 5 objetos en forma de archivo que contienen la información que se requiere para la emisión de un recibo.

II. Tipos de Objetos Internos:

- A. **Administración de Acceso.** Es el tipo de objeto controlador de acceso. Selecciona la información que es permisible para él para leer o escribir según su perfil.
- B. **Documento.** Cada elemento de información en la base de documentos de Lotus Notes, correspondiente a la información de un recibo en un periodo dado.
- C. **Recibo.** Es una copia de una instancia del tipo de objeto documento con la característica de que está plasmada en papel.

6.1.2 Esquema de Asociaciones de Objetos

La Figura 6.1 muestra las asociaciones entre los tipos de objetos identificados.

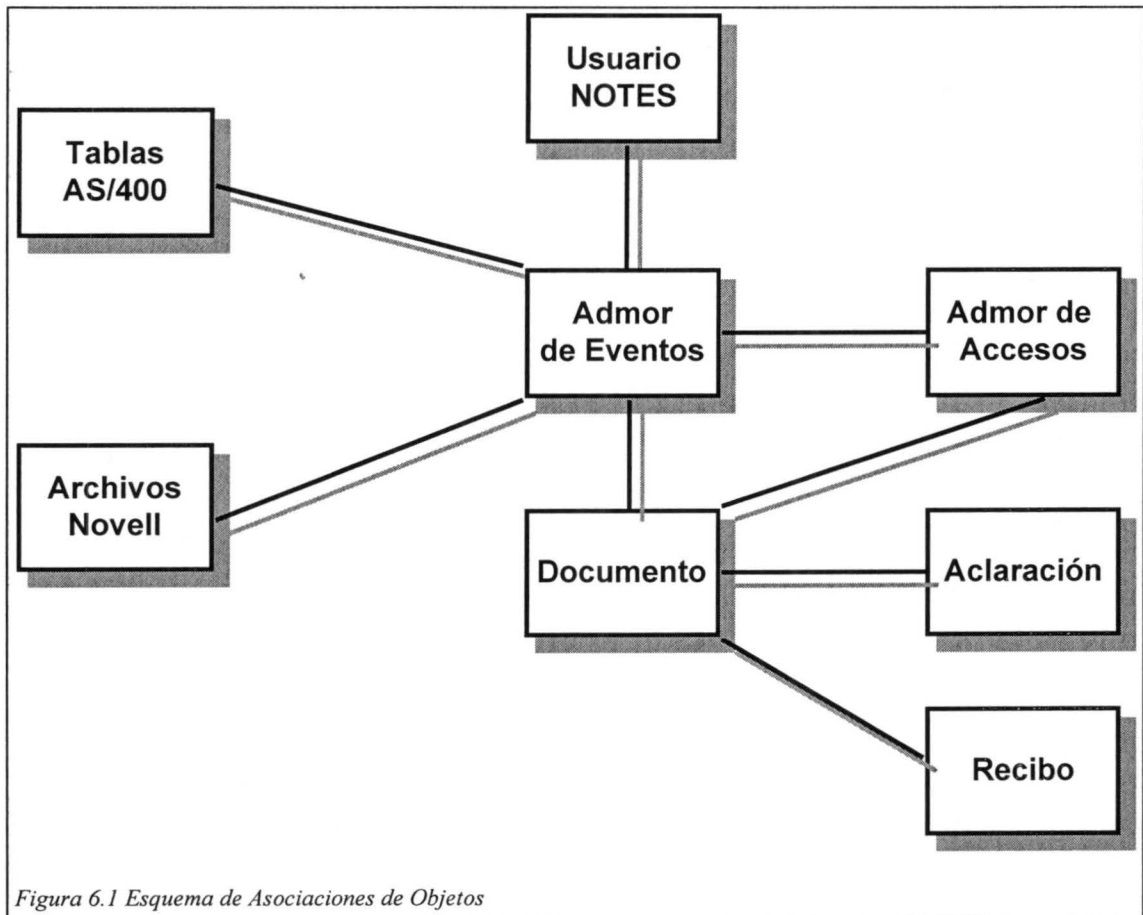
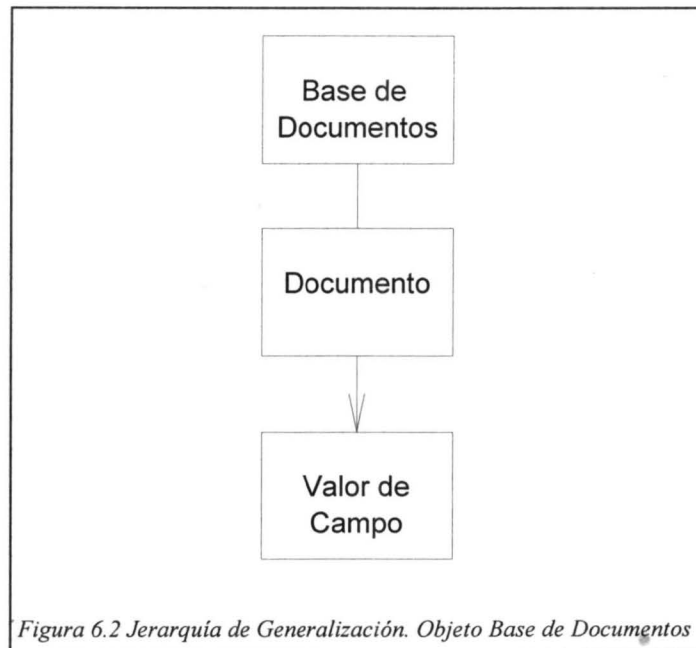


Figura 6.1 Esquema de Asociaciones de Objetos

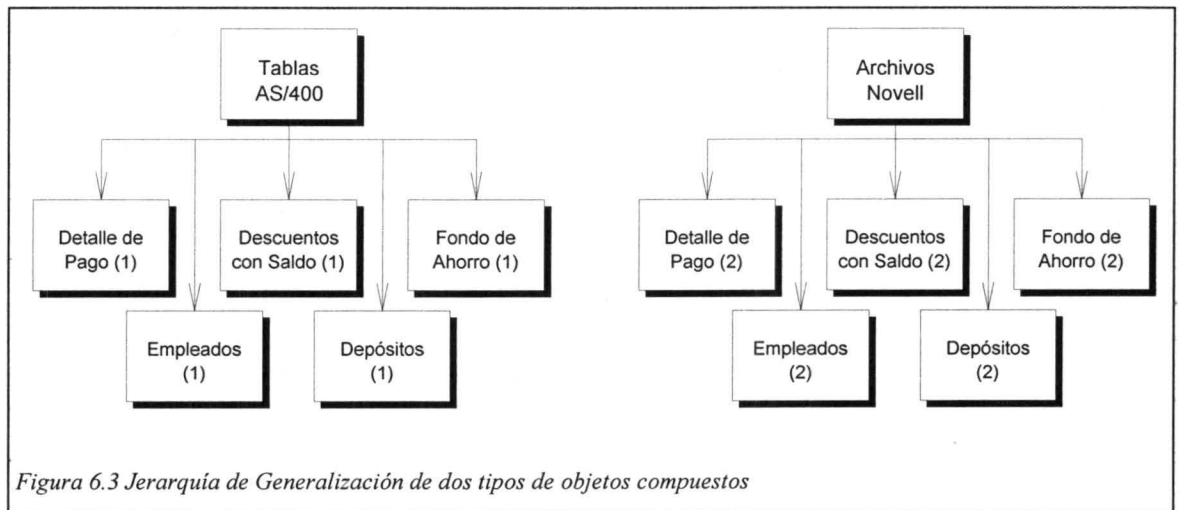
6.1.3 Jerarquías de Generalización de Objetos

La Figura 6.2 presenta la jerarquía de generalización del objeto *base de documentos*. Como puede observarse, éste es un objeto simple constituido por una jerarquía de 3 niveles.



Un par de objetos compuestos se muestran en la Figura 6.3. Como puede notarse, tanto el tipo de objeto *Tablas AS / 400* como el tipo de objeto *Archivos Novell* están compuestos cada uno de los objetos *Detalle de Pago*, *Descuentos con Saldo*, *Fondo de Ahorro*, *Empleados*, y *Depósitos*.

Ambos tipos de objetos podrán parecer a simple vista casi idénticos, de hecho, almacenan los mismos datos; la diferencia de ambos tipos de objetos radica en sus métodos, como se examinará más adelante. El número que se ha dispuesto entre paréntesis en cada una de las subclasses de la clase *Tablas AS/40* y de la clase *Archivos Novell* se han escrito a propósito para diferenciarlas una de otra.



6.2 Análisis del Comportamiento de Objetos (ACO)

En el Análisis del Comportamiento de Objetos se realizan esquemas de eventos. Estos esquemas muestran los eventos, la secuencia en que ocurren y cómo los eventos cambian el estado de los objetos [MARTIN, 94].

El ACO identifica la siguiente información:

1. Estados de los objetos
2. Transiciones de estado
3. Eventos que ocurren (cambios de estado de los objetos)
4. Operaciones que se llevan a cabo
5. Reglas de activación de los eventos

6.2.1 Definición del enfoque del análisis

Antes de comenzar con el esfuerzo del ACO, es necesario examinar lo que se desea y lo que no se desea modelar. En primer lugar hay que identificar el dominio del problema, esto es el universo particular de objetos que se desea modelar. El dominio, en este caso, se puede resumir en la siguiente sentencia: *“Proceso de verificación de pago de nómina, firma electrónica y retroalimentación con el área de Recursos Humanos”*.

En segundo lugar, se debe identificar el tipo de evento objetivo, el cual define el propósito del sistema que se está modelando. Junto a este tipo de evento

objetivo, es importante identificar también el tipo de evento inicial, el cual constituye el punto de partida del dominio.

Tipo de Evento Objetivo: “*Nómina verificada y confirmada*”. Donde por *nómina verificada* se refiere a la consulta del desglose del pago del empleado, y *nómina confirmada* significa que uno de dos eventos pudo haber ocurrido: *firma electrónica* o *solicitud de aclaración*.

Tipo de Evento Inicial: “*Nómina Ejecutada*”. Existe un disparador de reloj que cada día 15 del mes verifica si la información generada por la nómina ya está disponible.

6.2.2 Aclaración del Tipo de Evento

Asociado con el tipo de evento básico están los tipos de objetos preestado y postestado. El preestado de un evento especifica qué tipos de objetos se deben aplicar al objeto suyacente antes del evento. El postestado de un evento especifica qué tipos de objetos se deben aplicar al objeto subyacente después del evento [MARTIN. 94]. La Tabla 6.1 muestra la aclaración del tipo de evento para el sistema en análisis.

Preestado	Evento	Postestado
Información de Nómina importada	Nómina verificada	Registro Firma electrónica Correo con aclaración enviado

Tabla 6.1 Aclaración del tipo de evento

Una vez ubicado el tipo de evento básico del sistema podemos proceder a integrar los eventos relacionados con los tipos de objetos identificados que originarán el postestado deseado.

Las herramientas que se emplearán son el diagrama de transición de estados y el esquema general de eventos.

6.2.3 Identificación de los estados permisibles

Para poder definir lo que será el diagrama de eventos es necesario identificar los estados permisibles para cada objeto, y cuáles transiciones son válidas. En la Figura 6.4 se muestra esto.

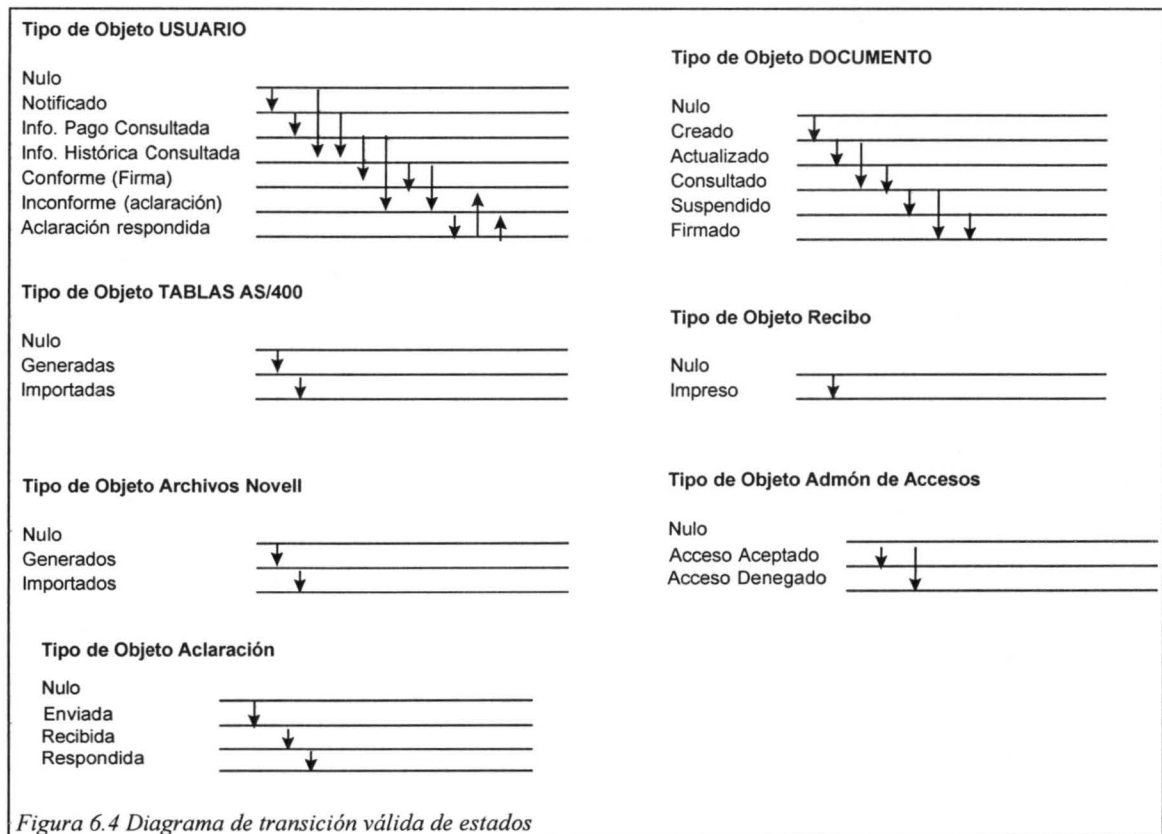


Figura 6.4 Diagrama de transición válida de estados

6.2.4 Definición de las condiciones de control

En este momento se requiere identificar la siguiente información:

1. Operaciones
2. Condiciones de control
3. Reglas de activación

Para producir los cambios de estado de la sección anterior se requiere de una *operación*. Una operación es una acción que modifica el estado de un objeto. Una operación puede ser interna o externa, según si se encuentra dentro o fuera del dominio especificado del sistema.

A menudo, una operación requerirá en un momento dado de ciertas condiciones para poder ser activada. Estas condiciones son llamadas *condiciones de control*. Las *reglas de activación* son funciones que asocian los objetos de un tipo de objeto con uno de otro tipo; se llaman así porque se requiere su presencia para que una operación pueda ser activada.

Las *operaciones*, las *condiciones de control* y las *reglas de activación* se representan mediante un diagrama llamado **esquema de eventos**. De la Figura 6.5 a la Figura 6.8 se muestra el diagrama correspondiente al HCA de Verificación de Pago de Nómina²².

El esquema de eventos se ha dividido en 4 procesos para su mejor visualización. La Figura 6.5 muestra la primera parte del proceso de Verificación de Nómina; esta parte comprende desde la *Importación de la Información de Nómina* hasta la *Notificación al Usuario*.

Como puede verse, existe una condición inicial que debe ser cumplida para que inicie el proceso. Esta condición valida la existencia de los siguientes casos:

1. Es día 15 y por lo tanto día de ejecución de la nómina ó
2. El administrador de Recursos Humanos ha solicitado la importación de los datos y
3. Se generaron las tablas con la información de la nómina JDE ó
4. Los archivos texto están disponibles en algún servidor de red

Más refinada, la cláusula condicional quedaría como:

²² La nomenclatura de este diagrama está basada en los estándares de diagramación que propone James Martin [MARTIN, 94], capítulo 9.

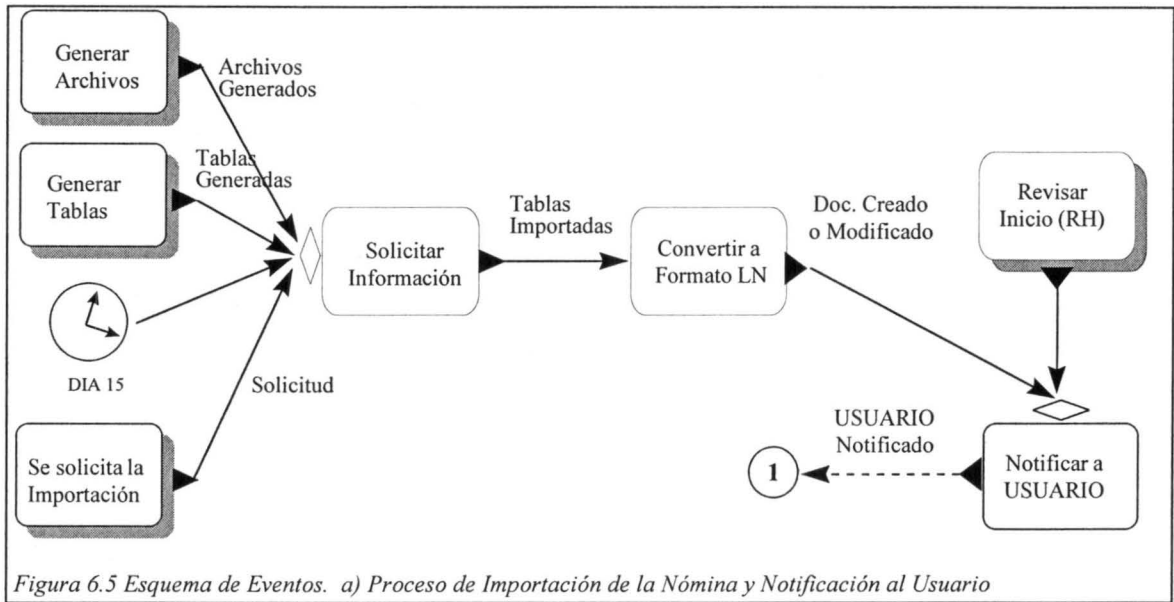


Figura 6.5 Esquema de Eventos. a) Proceso de Importación de la Nómina y Notificación al Usuario

SI [(Es día 15) O (Se solicita importación)] Y [(Tablas Generadas) O (Archivos Generados)] ENTONCES COMIENZA IMPORTACION.

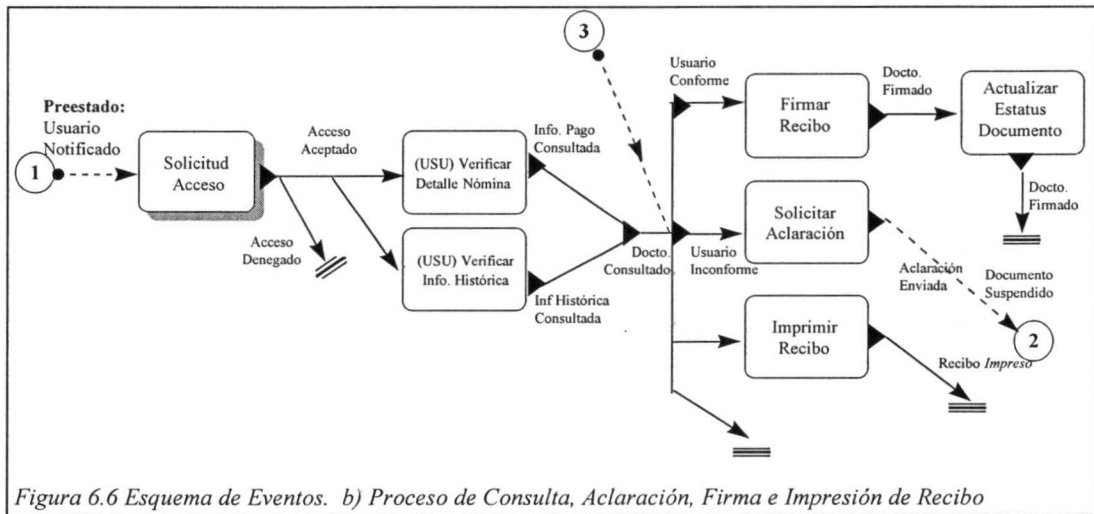
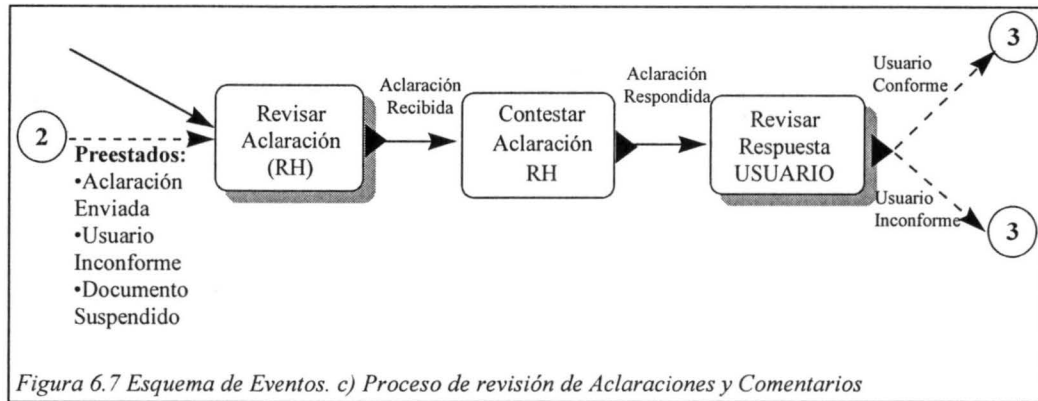


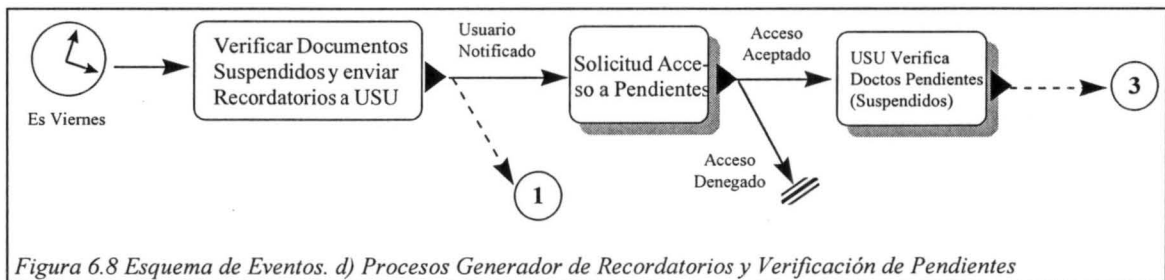
Figura 6.6 Esquema de Eventos. b) Proceso de Consulta, Aclaración, Firma e Impresión de Recibo

La Figura 6.6 presenta la parte del proceso que comprende la consulta del detalle de nómina por parte del empleado, la Firma electrónica en caso de estar conforme, la solicitud de aclaración en caso contrario, y la impresión del recibo.

La Figura 6.7 representa la continuación del esquema anterior en cuanto a solicitud de aclaraciones se refiere. Muestra los eventos relacionados con la revisión y retroalimentación de dicho tipo de evento.



En la Figura 6.8 se puede apreciar otro tipo de evento periódico. Cada semana se activará el proceso generador de recordatorios para enviar mensajes de correo electrónico a los usuarios que tengan compromisos (firmas) pendientes para recordarles.



7. DISEÑO DEL SISTEMA

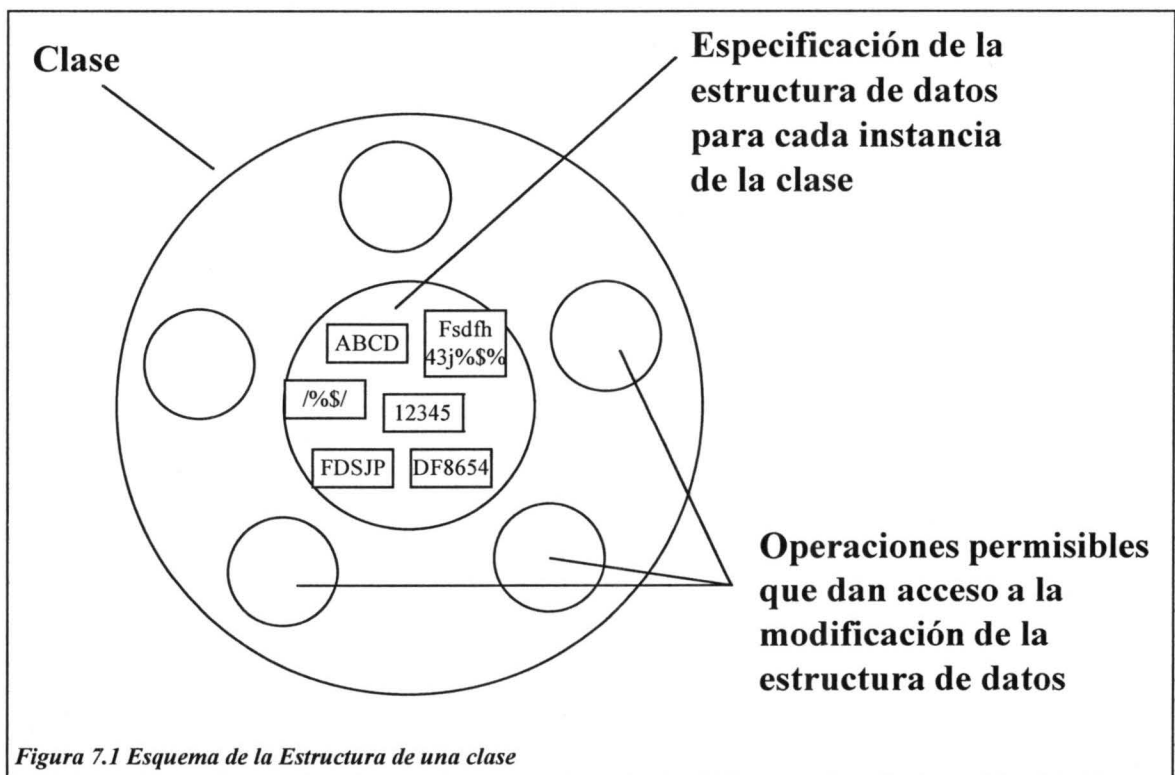
El contenido de este capítulo se apoya en los resultados del análisis arrojados en el capítulo anterior. El Diseño Orientado a Objetos que aquí se presenta está fundamentado en la metodología que propone James Martin [MARTIN, 94] y sustenta los Objetivos Personales No 4 y 5.

De la misma manera en que el análisis orientado a objetos contempla dos aspectos, el de la estructura y el del comportamiento de objetos (AEO, ACO), el diseño también se ocupa de ellos en lo que se denomina Diseño de la Estructura de Objetos (DEO) y el Diseño del Comportamiento de Objetos. Al igual que en el análisis, el DEO y el DCO están estrechamente relacionados, ya que los lenguajes de programación orientados a objetos (LPOO) tienen estructuras de datos y métodos combinadas dentro de las *clases*²³.

²³ El término clase se utiliza para denominar un tipo de objeto en un LPOO. "Clase es la implantación de un tipo de objeto. Especifica la estructura de datos y los métodos operativos que se aplican a cada uno de sus objetos" [MARTIN,94].

7.1 Diseño de la Estructura y el Comportamiento de Objetos a nivel Componentes

La Figura 7.1 muestra la estructura general de una clase. Como puede apreciarse la estructura de los datos y las operaciones se encuentran encapsulados dentro de la clase. “La especificación de cómo se llevan a cabo las funciones de una clase se llaman *métodos*” [MARTIN, 94]. De tal modo que los *métodos* son especificaciones en código de programación del procedimiento de una operación solicitada.



Una instancia de una clase es un objeto. Los datos y las operaciones que encapsula son especificados por su clase. El objeto almacena sus propios datos, los cuales sólo pueden ser modificados a través de sus métodos.

7.2 Diagramas de DEO y DCO de los Componentes del Sistema

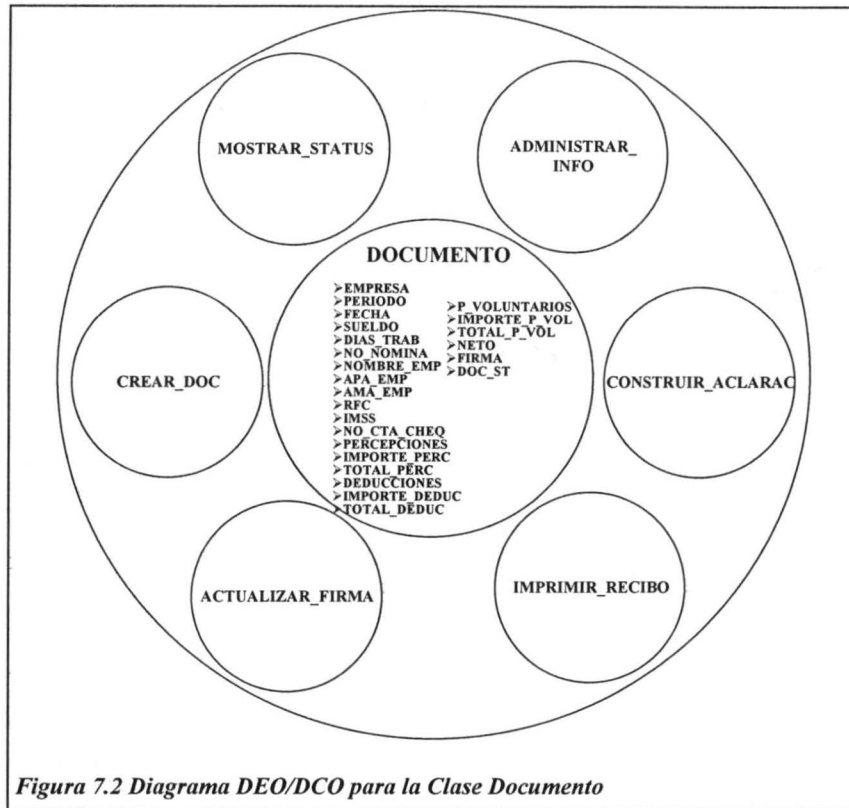
Durante el análisis de la estructura y comportamiento de objetos se definió el “Qué” de la solución al problema. En el diseño se pretende definir el “Cómo”. Aunque no existe una manera universalmente aceptada, es importante que durante el diseño se especifiquen los atributos y los métodos de los tipos de objeto internos al sistema identificados durante el análisis.

En el diseño, los diagramas de la estructura de objetos deben reflejar toda la funcionalidad descrita en los diagramas del AEO. Los diagramas del comportamiento deben soportar todas las operaciones definidas en el esquema de eventos (ACO).

Para su mejor visualización, y para reafirmar el concepto de encapsulación, esta sección presenta en primer lugar los diagramas correspondientes al diseño de la estructura y comportamiento de objetos en forma conjunta para cada clase y en seguida una breve descripción de los métodos y atributos que la componen.

7.2.1 Clase DOCUMENTO

La clase **DOCUMENTO** se refiere al objeto principal del sistema, el cual contiene los datos referentes a la información del pago de la nómina que serán consultados por los empleados. Los métodos de esta clase se encargan de la administración de la información contenida en el núcleo de la clase. La Figura 7.2 ilustra el diagrama correspondiente a esa clase, la Tabla 7.1 explica sus atributos y la Tabla 7.2 sus métodos.



ATRIBUTO	DESCRIPCIÓN
EMPRESA	Compañía a la que pertenece el empleado
PERIODO	Período de Nómina que se está reportando
FECHA	Fecha de Pago
SUELDO	Sueldo del Empleado
DIAS_TRAB	No. de días laborados en el período
NO_NOMINA	Identificador de Nómina del Empleado
NOMBRE_EMP	Nombre del Empleado
APA_EMP	Apellido Paterno
AMA_EMP	Apellido Materno
RFC	Registro Federal de Causantes
IMSS	No. de registro de Seguro Social
NO_CTA_CHEQ	No. de cuenta de cheques
PERCEPCIONES	Conceptos de percepción
IMPORTE_PERC	Importe de percepción
TOTAL_PERC	Total de percepciones
DEDUCCIONES	Conceptos de deducción
IMPORTE_DEDUC	Importe de deducción
TOTAL_DEDUC	Total de deducciones

P_VOLUNTARIOS	Conceptos de Pagos Voluntarios
IMPORTE_P_VOL	Importe de Pagos Voluntarios
TOTAL_P_VOL	Total de Pagos Voluntarios
NETO	Cantidad Neta a Recibir
FIRMA	Indicador de que el documento fue electrónicamente firmado
DOC_ST	Estado actual del documento

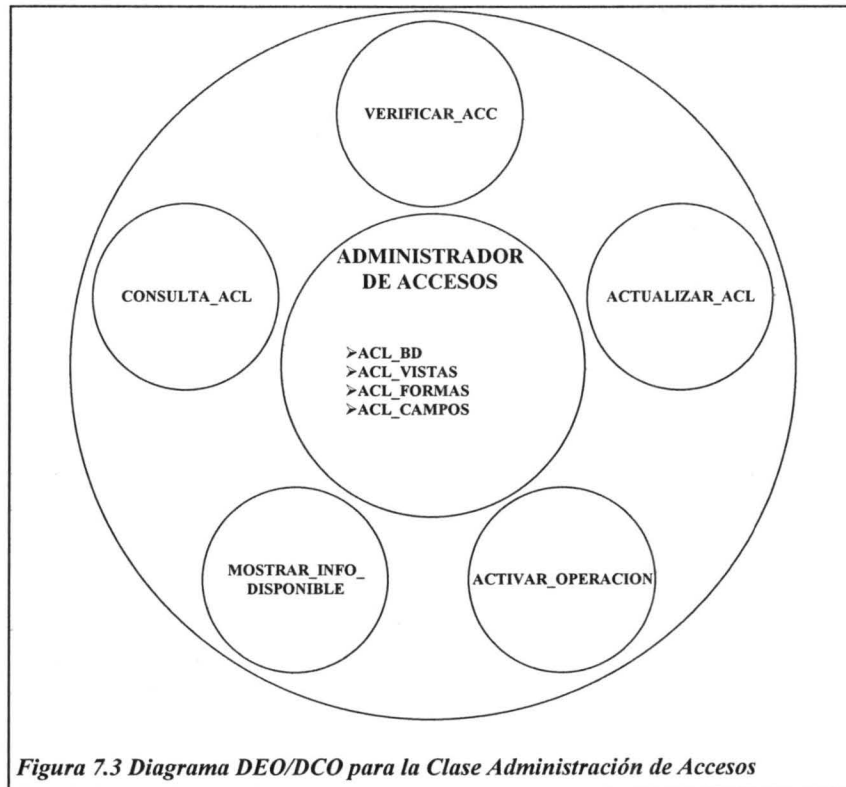
Tabla 7.1 Atributos de la Clase Documento

MÉTODO	DESCRIPCIÓN
CREAR_DOC	Crea un documento nuevo con la información disponible.
ADMINISTRAR_INFO	Este método controla el acceso a la información del documento
MOSTRAR_STATUS	Este método revela el status del documento; un documento puede estar firmado, en proceso de aclaración, o pendiente (no firmado + sin aclaración)
ACTUALIZAR_FIRMA	Actualiza en el documento el registro de conformidad del empleado .
IMPRIMIR_INFO	Envía un mensaje a objeto RECIBO para que el desglose de pago sea impreso.
CONSTRUIR_ACLARAC	Envía un mensaje de creación al tipo de objeto ACLARACIÓN

Tabla 7.2 Métodos de la Clase Documento

7.2.2 Clase ADMINISTRADOR ACCESOS

En un sentido figurado el tipo de objeto **ADMINISTRADOR ACCESOS** es el ente que se encarga de mostrar u ocultar la información al usuario; esto es, según el rol y el acceso definidos para cada usuario, éste podrá sólo ver la información que le compete. De la misma forma, el tipo de objeto **ADMINISTRADOR ACCESOS** puede restringir el acceso a sólo lectura, o permitir también la escritura (para el caso de la firma y las aclaraciones). Cabe señalar que este tipo de objeto contempla la administración del acceso a nivel base de datos, nivel vista, nivel forma y nivel campo. La Figura 7.3 ilustra el esquema DEO / DCO de la clase correspondiente a este tipo de objeto.



La Tabla 7.3 y la Tabla 7.4 muestran respectivamente los atributos y métodos del DEO / DCO de la Clase **ADMINISTRACIÓN DE ACCESOS**.

ATRIBUTO	DESCRIPCIÓN
ACL_BD	Lista de Control de Acceso a la Base de Datos
ACL_VISTAS	Lista de Roles para el acceso a nivel vista
ACL_FORMAS	Lista de Roles para el acceso a nivel forma
ACL_CAMPOS	Lista de Nombres para el acceso a nivel campo

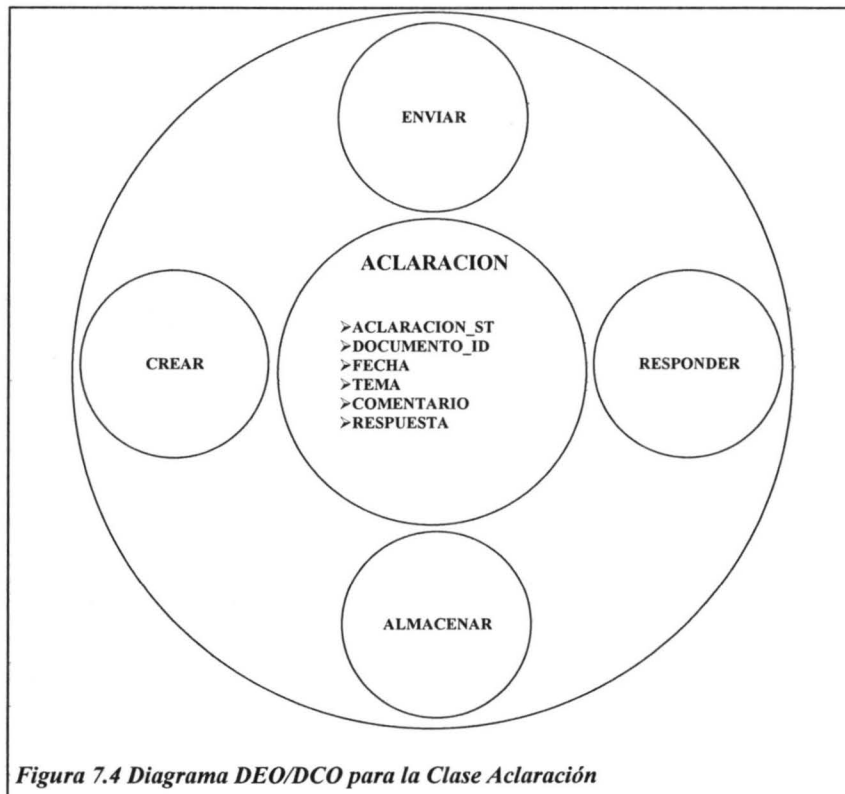
Tabla 7.3 Atributos de la Clase Administración de Accesos

MÉTODO	DESCRIPCIÓN
REVISAR_ACC	Verifica que el acceso solicitado a través del objeto ADMINISTRADOR DE EVENTOS sea válido.
MOSTRAR_INFO_DISPONIBLE	En función del acceso que fue otorgado selecciona las vistas, documentos y secciones que mostrarán al usuario.
ACTIVAR_OPERACION	En función del acceso otorgado, así como el estímulo activado por la entidad externa a través del ADMINISTRADOR_DE_EVENTOS activa los métodos adecuados de la clase DOCUMENTO.
CONSULTA_ACL	Ante una solicitud, devuelve la información contenida en el ACL (nombres, identificadores y roles).

Tabla 7.4 Métodos de la Clase Administración de Accesos

7.2.3 Clase ACLARACIÓN

La clase **ACLARACIÓN** encapsula todas las operaciones que se requieren para crear, enviar, consultar y responder una aclaración. La Figura 7.4 muestra su esquema de DEO / DCO.



La Tabla 7.5 y la Tabla 7.6 muestran los atributos y métodos de esta clase.

ATRIBUTO	DESCRIPCIÓN
ACLARACION_ST	Estado de la aclaración
DOCUMENTO_ID	Identificador único del documento con el que está relacionado
FECHA	Fecha de creación de la aclaración
TEMA	Resumen en una línea del detalle de la aclaración
COMENTARIO	Detalle de la aclaración
RESPUESTA	Detalle de la respuesta

Tabla 7.5 Atributos de la Clase Aclaración

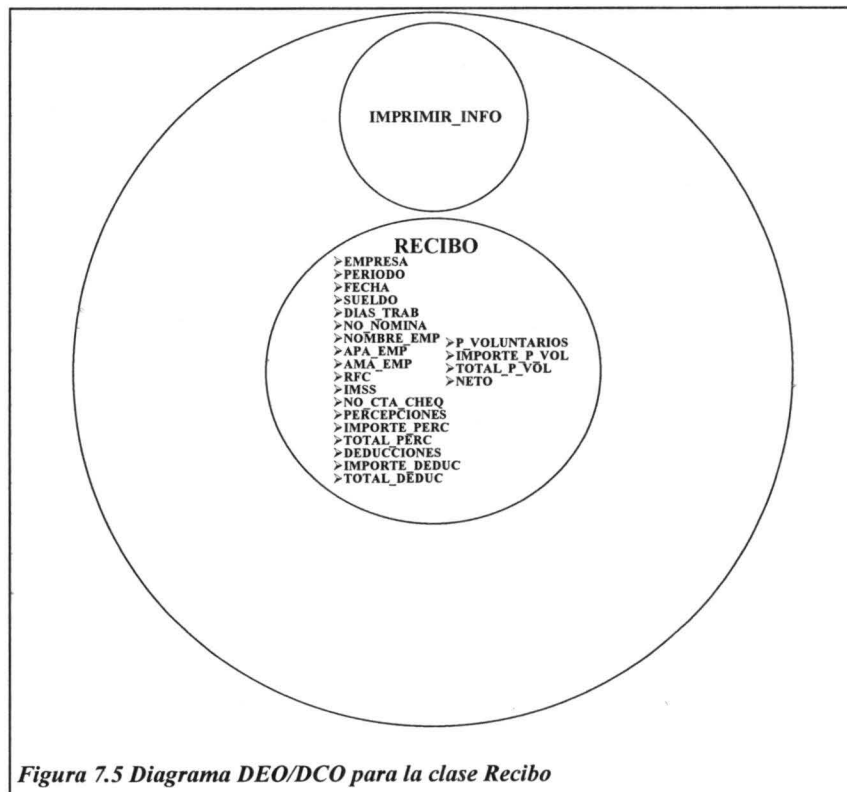
MÉTODO	DESCRIPCIÓN
CREAR	Crea una aclaración nueva
ENVIAR	Envía una aclaración creada o modificada

RESPONDER	Devuelve una aclaración respondida a su emisor original
ALMACENAR	Guarda una aclaración para su posterior envío

Tabla 7.6 Métodos de la Clase Aclaración

7.2.4 Clase RECIBO

El recibo es la impresión de la información de pago de nómina en papel. La clase **RECIBO** se encarga de esta operación. La Figura 7.5 ilustra el diagrama DEO/DCO para esta clase.



El único método que pertenece a esta clase es *IMPRIMIR_INFO* y se encarga de imprimir los datos almacenados en el objeto.

La Tabla 7.7 describe los atributos de esta clase, nótese que aunque es muy similar a la Tabla 7.1 (Atributos de la Clase Documento) se han omitido algunos atributos que no son relevantes.

ATRIBUTO	DESCRIPCIÓN
----------	-------------

EMPRESA	Compañía a la que pertenece el empleado
PERIODO	Período de Nómina que se está reportando
FECHA	Fecha de Pago
SUELDO	Sueldo del Empleado
DIAS_TRAB	No. de días laborados en el período
NO_NOMINA	Identificador de Nómina del Empleado
NOMBRE_EMP	Nombre del Empleado
APA_EMP	Apellido Paterno
AMA_EMP	Apellido Materno
RFC	Registro Federal de Causantes
IMSS	No. de registro de Seguro Social
NO_CTA_CHEQ	No. de cuenta de cheques
PERCEPCIONES	Conceptos de percepción
IMPORTE_PERC	Importe de percepción
TOTAL_PERC	Total de percepciones
DEDUCCIONES	Conceptos de deducción
IMPORTE_DEDUC	Importe de deducción
TOTAL_DEDUC	Total de deducciones
P_VOLUNTARIOS	Conceptos de Pagos Voluntarios
IMPORTE_P_VOL	Importe de Pagos Voluntarios
TOTAL_P_VOL	Total de Pagos Voluntarios
NETO	Cantidad Neta a Recibir

Tabla 7.7 Atributos de la clase Recibo

7.2.5 Elemento RELOJ

Existe un agente adicional que se debe tomar en consideración para el diseño del sistema. Este agente es el responsable de administrar los procesos que son periódicos y se denomina **RELOJ**. Aunque algunos autores preferirían no definir el reloj como un tipo de objeto por claridad en este sistema se define como una especie de “*clase*” con la particularidad de que no tiene atributos explícitos. El **RELOJ** sirve como activador de acciones dentro del sistema, se comunica con los métodos de las clases cuyas funciones son periódicas. La Figura 7.6 ilustra la representación del agente en cuestión.

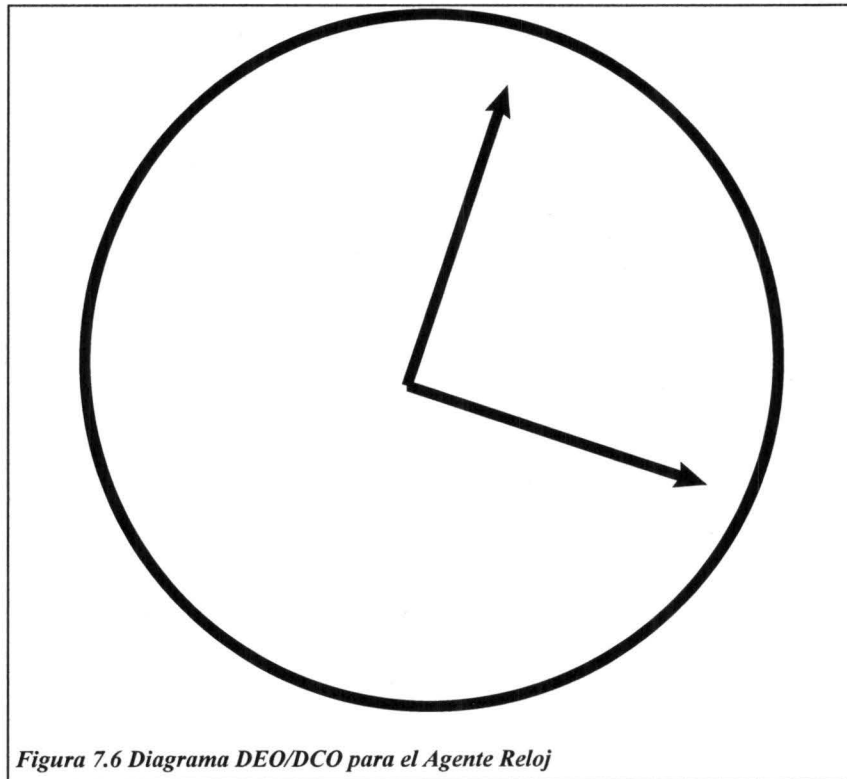
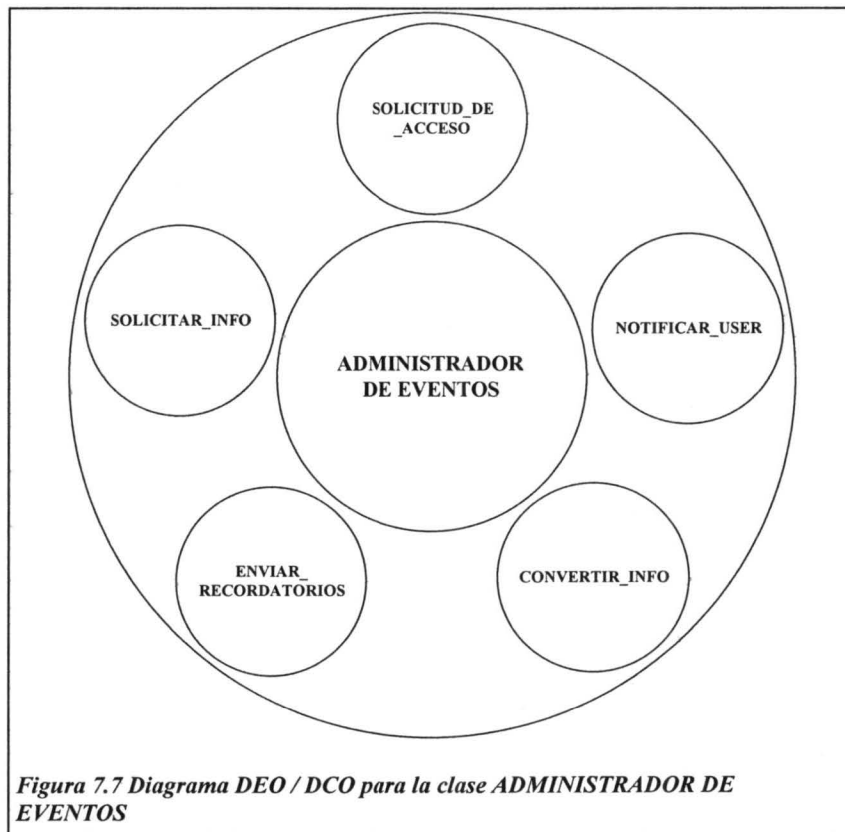


Figura 7.6 Diagrama DEO/DCO para el Agente Reloj

7.2.6 Clase ADMINISTRADOR DE EVENTOS

El último elemento, pero no por eso menos importante, es el **ADMINISTRADOR DE EVENTOS**. Todo lo que sucede dentro del sistema tiene que pasar primero por este tipo de objeto. Podría decirse que es un ente “abstracto” que colecciona métodos los cuales son activados por estímulos ambientales; estos estímulos pueden ir desde peticiones explícitas del usuario del sistema hasta activaciones de RELOJ.

En la Figura 7.7 se ilustra el diagrama correspondiente a la clase ADMINISTRADOR DE EVENTOS.



Podríamos imaginar el ADMINISTRADOR DE EVENTOS como un agente de tránsito. Frente a él pasan todos los automóviles en un cruce y su función es facilitar el flujo de ellos; pero si alguno de ellos intentara infringir las reglas, éste tomará las acciones debidas, o en su caso, impedirá que el conductor haga algo indebido. De la misma manera el ADMINISTRADOR DE EVENTOS monitorea todos los eventos del sistema y se encarga de facilitar que ocurran de la manera más natural posible; si algo como la solicitud de un acceso no autorizado sucede, lo impide. Dada la naturaleza transitiva del ADMINISTRADOR DE EVENTOS, no tiene atributos propios, pero sí métodos que se encargan de canalizar la información que manipula a las entidades adecuadas.

La Tabla 7.8 ilustra los métodos del ADMINISTRADOR DE EVENTOS.

MÉTODOS	DESCRIPCIÓN
SOLICITUD_DE_ACCESO	Este método es activado por el usuario. Cualquier acceso a las acciones del sistema se realizará a través de este método.
ENVIAR_RECORDATORIO	Este método es activado cada semana para verificar cuáles empleados tienen firmas de documentos pendientes y les envía un recordatorio.
SOLICITAR_INFORMACIÓN	Cada mes establece la comunicación con el IBM AS/400. Consulta los datos de la base de datos relacional definida, o en su caso, la comunicación con el servidor de red para realizar la importación de los archivos .TXT
CONVERTIR_INFORMACIÓN	Una vez que la información es importada es requerido manipularla para darle el formato correcto. Es distinto el procedimiento para la conversión de la información proveniente del AS/400 de la proveniente del servidor de red; el método deberá ser lo suficientemente inteligente para manejar una u otra forma.
NOTIFICAR_USER	Envía un mensaje personalizado a través del correo electrónico para notificar al usuario que la información ya está disponible. Además del mensaje se envía un enlace para agilizar el acceso a la herramienta.

Tabla 7.8 Métodos relacionados con la clase ADMINISTRADOR DE EVENTOS

7.3 Resumen del DEO / DCO para los componentes del sistema

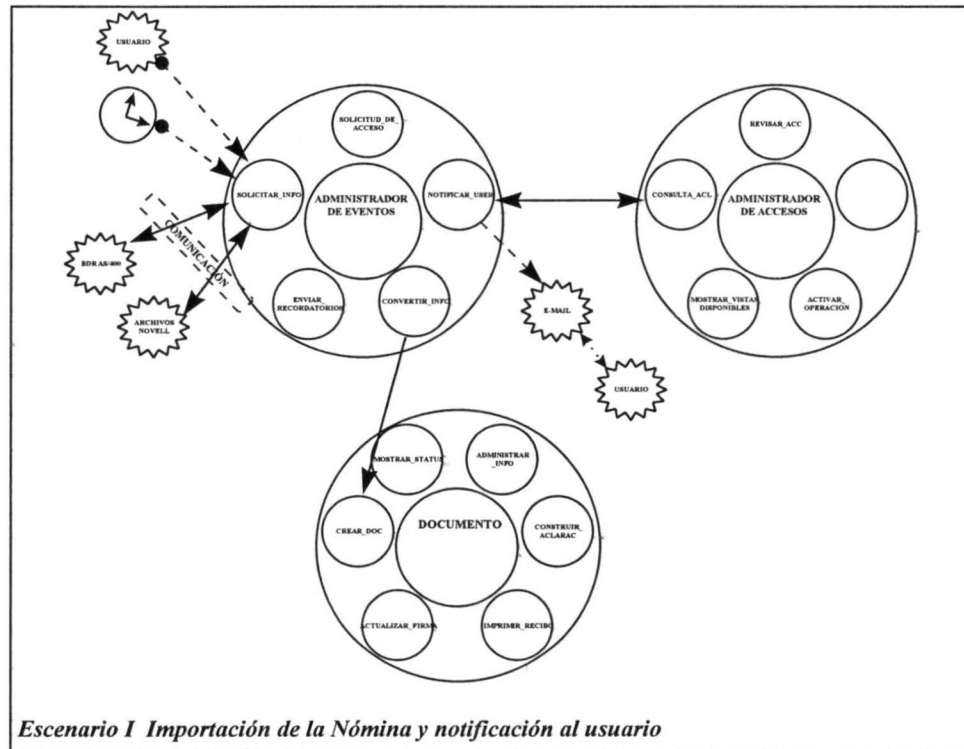
Hasta ahora se han definido los métodos y atributos que definen cada clase de objeto por separado; en la Tabla 7.9 se resume estas características en un sólo cuadro global para su mejor visualización.

CLASE	ATRIBUTOS	MÉTODOS
Documento	EMPRESA, PERIODO, FECHA, SUELDO, DIAS_TRAB, NO_NOMINA, NOMBRE_EMP, APA_EMP, AMA_EMP, RFC, IMSS, NO_CTA_CHEQ, PERCEPCIONES, IMPORTE_PERC, TOTAL_PERC, DEDUCCIONES, IMPORTE_DEDUC, TOTAL_DEDUC, P_VOLUNTARIOS, IMPORTE_P_VOL, TOTAL_P_VOL, NETO, FIRMA, DOC_ST	MOSTRAR_STATUS ADMINISTRAR_INFO CREAR_DOC ACTUALIZAR_FIRMA IMPRIMIR_INFO CONSTRUIR_ACLARAC
Admón_Accesos	ACL_BD, ACL_VISTAS, ACL_FORMAS, ACL_CAMPOS	REVISAR_ACC CONSULTA_ACL MOSTRAR_INFO_DISPONIBL ACTIVAR_OPERACIÓN
Sol_Aclaración	ACLARACION_ST, DOCUMENTO_ID, FECHA, COMENTARIO, RESPUESTA	CREAR, ENVIAR, ALMACENAR, RESPONDER
Recibo	EMPRESA, PERIODO, FECHA, SUELDO, DIAS_TRAB, NO_NOMINA, NOMBRE_EMP, APA_EMP, AMA_EMP, RFC, IMSS, NO_CTA_CHEQ, PERCEPCIONES, IMPORTE_PERC, TOTAL_PERC, DEDUCCIONES, IMPORTE_DEDUC, TOTAL_DEDUC, P_VOLUNTARIOS, IMPORTE_P_VOL, TOTAL_P_VOL, NETO	IMPRIMIR_INFO
[RELOJ]	[No tiene atributos propios]	ACTIVAR(<i>método, objeto</i>)
ADMINISTRADOR DE EVENTOS	[No tiene atributos propios]	SOLICITUD_DE_ACCESO ENVIAR_RECORDATORIO NOTIFICAR_USER SOLICITAR_INFO CONVERTIR_INFO

Tabla 7.9 Resumen del DEO / DCO de los componentes del sistema

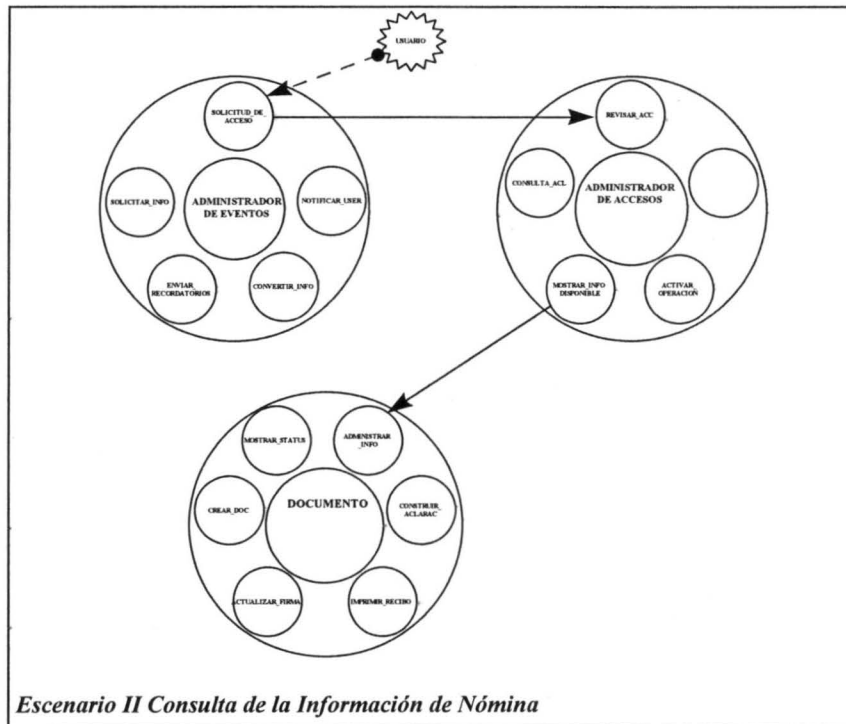
7.4 Diseño de la Estructura y Comportamiento de Objetos a nivel Sistema

Una vez establecido el diseño de los componentes del sistema, se precisa conocer los detalles de la interacción de éstos. En este apartado se mostrará al lector a través de escenarios *cómo* es que estos componentes se relacionan para completar los objetivos del sistema. La razón de utilizar escenarios es facilitar la comprensión de las funciones representadas en los diagramas.



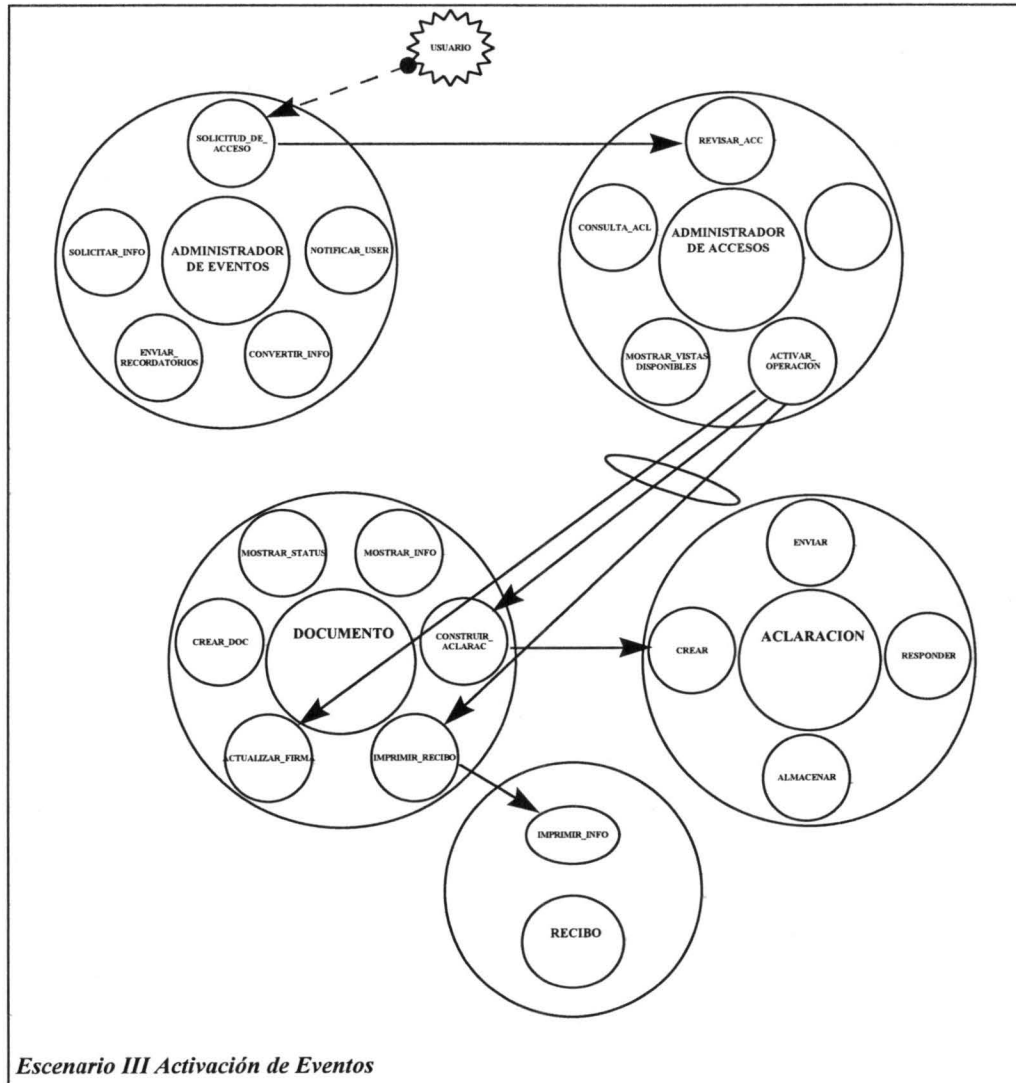
El Escenario I describe los métodos empleados por las clases ADMINISTRADOR DE EVENTOS y DOCUMENTO para la realización del proceso de importación de la nómina y depósito de la información en la base de documentos del sistema. Además, muestra los métodos que intervienen en la notificación al usuario ; el método NOTIFICAR_USER obtiene del método ACTUALIZAR_ACL la lista de los usuarios del sistema a los cuales se les notificará vía correo electrónico. y aquellos que puede activar para alterar el sistema. Nótese cómo la interacción de los objetos internos al sistema con entidades externas se denota con una flecha punteada, a diferencia de la interacción entre dos objetos que son internos; la razón consiste en diferenciar cuándo el control lo lleva el sistema y cuándo el control es transferido a la entidad externa. La flecha continua denota la activación de un método a otro a

través de *mensajes*. Para mayor información sobre la interacción del sistema con entidades externas consúltese el APÉNDICE A.

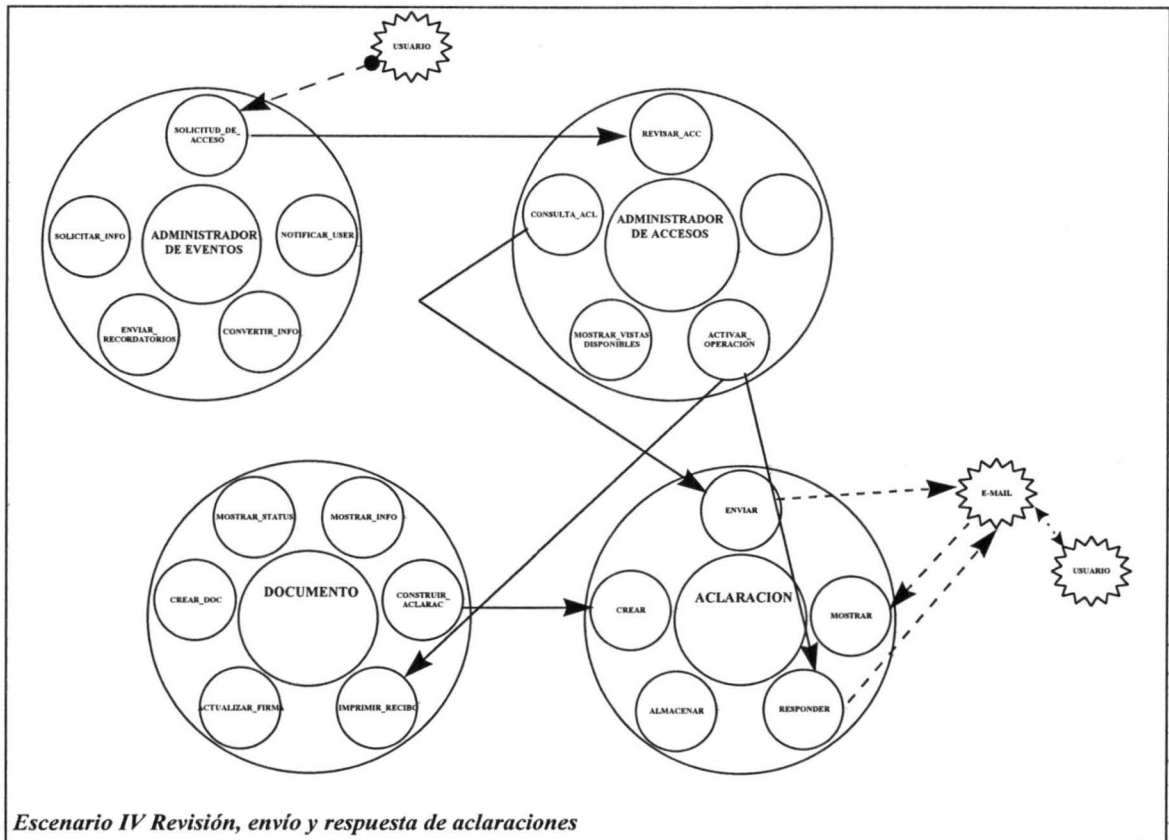


El Escenario II muestra el proceso interno de comunicación de las clases para mostrar la información adecuada a la persona adecuada. El método selecciona qué información mostrará al usuario y otorgará los privilegios apropiados según su perfil.

El Escenario III presenta el proceso de activación de eventos relacionados con la clase DOCUMENTO. El anillo colocado alrededor de las flechas salientes indica que sólo una activación se realiza a la vez.

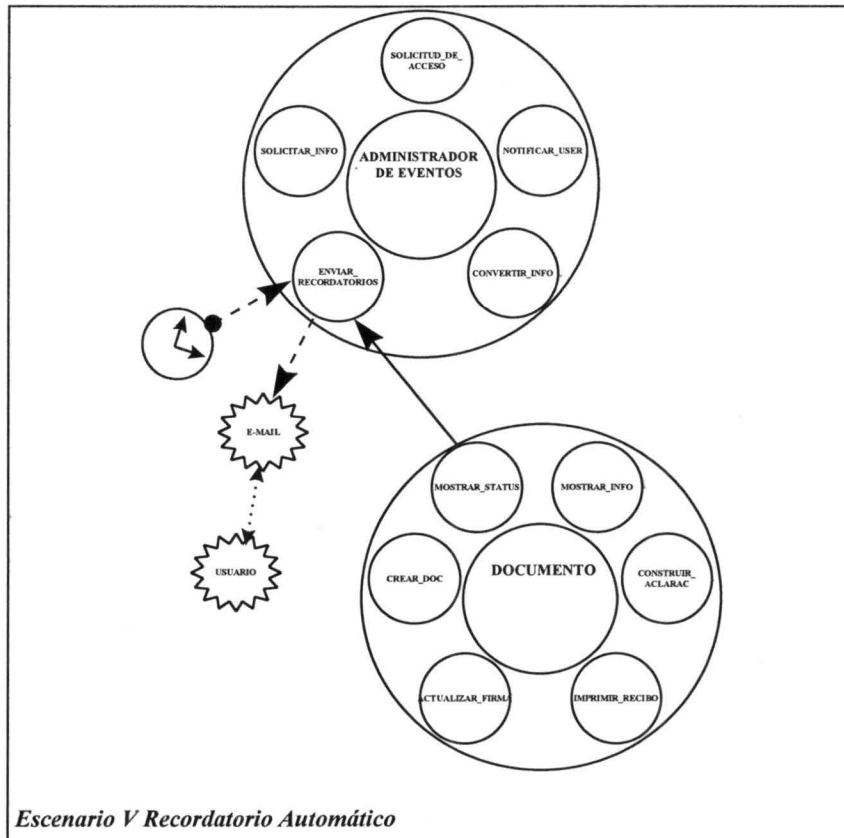


A través de la activación de una operación, el usuario podrá emitir una firma electrónica, imprimir la información o bien construir una solicitud de aclaración, la cual según el diagrama presentado en el Escenario IV podrá ser remitida al departamento de Recursos Humanos.



Para que una Solicitud de Aclaración pueda ser enviada, el método ENVIAR obtendrá primero la lista de las personas que poseen el *ROL* de "Revisor" de RH y luego enviará un enlace esta solicitud a través del correo electrónico. La(s) persona(s) designada(s) para atender las inquietudes de los empleados podrá(n) entonces acceder la herramienta para consultar las aclaraciones y responderlas al usuario original.

Finalmente, el proceso de recordatorios automáticos es representado en el Escenario V; como puede apreciarse, el RELOJ activa el método ENVIAR_RECORDATORIOS del ADMINISTRADOR DE EVENTOS el cual mediante un mensaje al método MOSTRAR_STATUS obtiene la lista de usuarios con compromisos pendientes y les envía un mensaje a través del correo electrónico.



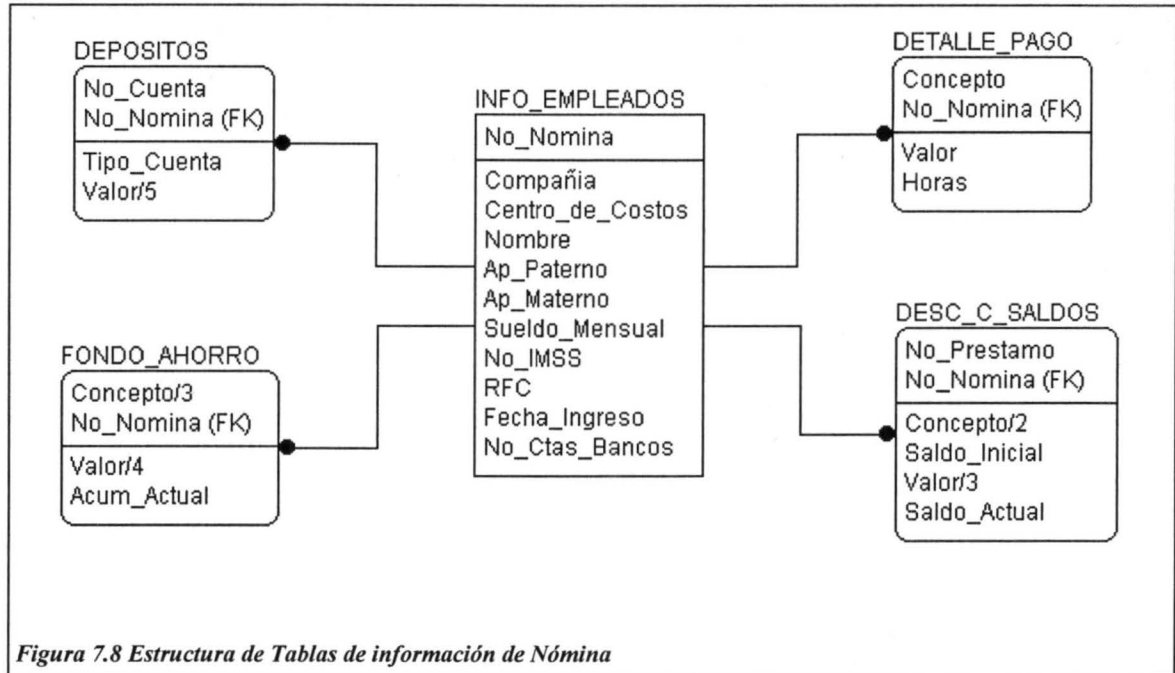
7.5 Diseño de la Base de Datos Relacional (BDR)

Como se comentó en los objetivos particulares, la importación de la información de nómina se efectuará a elección del usuario de una de las siguientes formas:

- a) Desde una BDR en el AS / 400
- b) Desde un archivo tipo texto en un servidor Novell

Para este efecto dos estándares deben ser propuestos: uno para la base de datos relacional, y otro para los archivos con el formato texto.

Por simplicidad se propone que ambos estándares coincidan en su diseño, sin embargo es importante destacar que aunque a simple vista pueda parecer que en esencia son lo mismo, los detalles estructurales y de manipulación de los datos difieren enormemente. La Figura 7.8 muestra la estructura de las tablas propuestas para los dos casos expuestos.



Además de estas tablas se requerirá de disponer de los archivos F069116, F069066 y F06210 de la Nómina de JDE para la obtención de la descripción de los conceptos y para el cálculo del periodo de pago.

7.6 Diseño de la Base de Documentos

Debido a que Lotus Notes administra documentos, el término “Base de documentos” es más adecuado para referirse al acervo de información que administra. Elaborando un análisis comparativo entre lo que es una *base de datos* y una *base de documentos* se obtiene la información que aparece en la Tabla 7.10.

EQUIVALENCIA	
BASE DE DATOS	BASE DE DOCUMENTOS
Tipo de Relación (tabla)	Vista (no exactamente)
Registros (renglones)	Documentos
Atributos (columnas)	Campos uni y multivalores

Tabla 7.10 Equivalencias aproximadas entre una base de datos y una base de documentos

A pesar de las equivalencias aproximadas, modelar una base de documentos a partir de una base de datos no siempre se realizará en forma directa; por ejemplo, los Tipo de Relaciones expresados en la Figura 7.8 podrían en un momento dado convertirse cada uno directamente en un conjunto de

documentos, sin embargo esto tendría la desventaja de que a la hora de la consulta se tendría que estar buscando información de diferentes documentos para desplegarlos en uno sólo, lo cual disminuiría el rendimiento del sistema a la hora de la ejecución. La alternativa que se presenta es crear el documento tal cual como se mostrará al usuario durante la ejecución, sacrificando un poco el tiempo de la importación.

Una vez tomada en cuenta esta consideración y los otros elementos especificados en el diseño del sistema, el diseño de la base de documentos resulta como se muestra en el Apéndice B de este documento.

7.7 Diseño de las vistas

La Figura 7.9 presenta las vistas conceptualizadas en el diseño y sus categorías. Normalmente a cada vista se le asocian una o más formas.

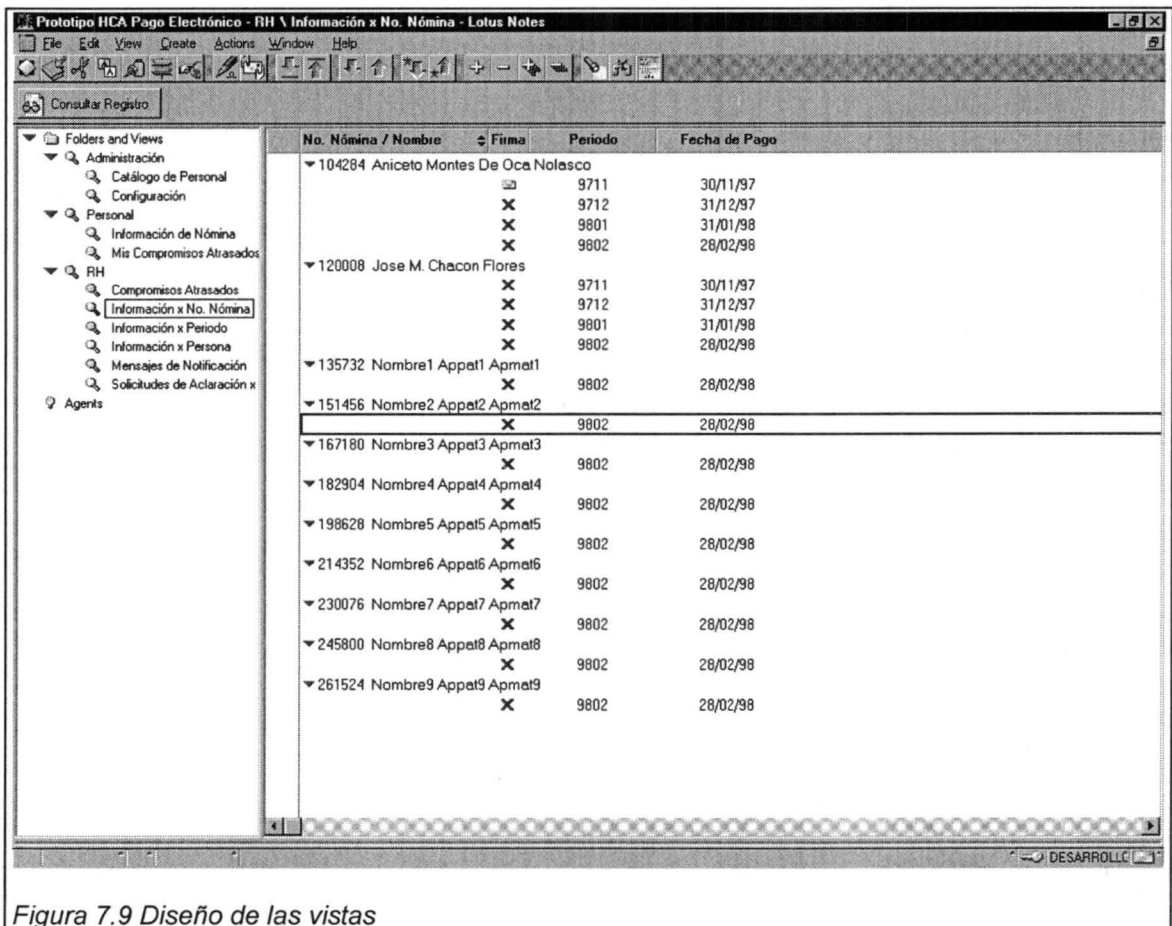


Figura 7.9 Diseño de las vistas

Existen 3 categorías de vistas:

1. Administración
2. Personales
3. Recursos Humanos

7.7.1 Vistas de Administración

Las vistas de *Administración*, como su nombre lo indica, están destinadas a facilitar la administración del sistema. Se diseñaron dos vistas para este efecto:

- a) Catálogo de Personal
- b) Configuración del Sistema

El *Catálogo de Personal* es un directorio que relaciona los números de nómina de los empleados con sus respectivos nombres identificadores de Lotus Notes. Estos nombres constituyen un identificador único mediante el cual el Lotus Notes puede interactuar con el usuario a través del correo electrónico o las aplicaciones. La justificación para la creación de esta vista consiste en que es necesario establecer una relación entre la información de nómina y la entidad Usuario Notes descrita en la sección del Análisis de la Estructura de Objetos. La vista *Catálogo de Personal* desplegará en columnas tanto el Número de Nómina como el Identificador *Nombre Notes*.

La vista *Configuración del Sistema* debe contener únicamente 1 documento. Este documento establece los parámetros bajo los cuales el sistema ejecutará su trabajo. Parámetros como la ubicación de la información fuente y la fecha de la última importación aparecen en este documento. La vista mostrará la fecha de la última modificación del documento, así como la fuente y ruta de acceso de la información de nómina tal y como se ilustra en la Figura 7.10.

7.7.2 Vistas de Tipo Personal

Existen dos vistas que caen en esta categoría. La primer vista desplegará al usuario que tenga el rol de empleado todos los documentos que poseen la Información de Nómina de los cuales su Nombre Notes aparezca almacenado. En otras palabras, el empleado que no posee otro rol adicional podrá acceder

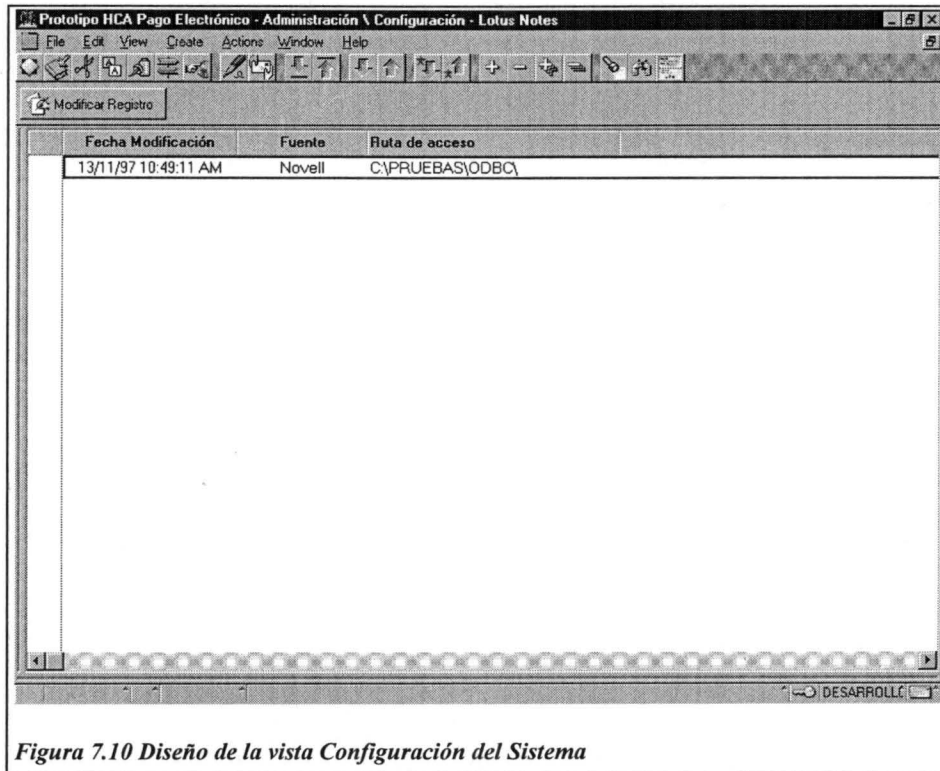


Figura 7.10 Diseño de la vista Configuración del Sistema

su propia Información de Nómina a través de esta vista. De cada documento se desplegará el Periodo de pago, la fecha de pago, un indicador de firma y la fecha de la firma, en caso de existir.

La segunda vista es similar a la anterior, sólo que mostrará únicamente aquellos documentos cuyo indicador de firma no esté activado; esto con la finalidad de que el usuario pueda localizar rápidamente cuáles son sus compromisos de firma atrasados.

7.7.3 Vistas de Tipo RH

Este tipo de vistas son sólo accesibles por el personal que posee el rol de *Recursos Humanos* o el rol de *Observador*. La finalidad de diseñar vistas que caen dentro de esta categoría es proveer a los administradores la capacidad de monitorear los sueldos de los empleados, así como los compromisos de firmas atrasadas.

Tres de las vistas de Tipo RH muestran la totalidad de los documentos que contienen la Información de Nómina de todos los empleados ordenados por Periodo, Número de Nómina y Nombre del Empleado, respectivamente.

La vista llamada *Compromisos Atrasados*, de forma muy similar a la de Tipo *Personal*, desplegará ordenados por periodo, todos los nombres y documentos de aquellos empleados cuyos documentos estén pendientes de firmar. De esta manera el control sobre los compromisos pendientes será mucho más sencillo.

La vista *Mensajes de Notificación* es un catálogo de todos los mensajes que empleará el sistema para el envío de notificaciones y recordatorios a través del correo electrónico. De cada uno de ellos la vista mostrará su identificador y un breve abstracto del asunto.

Existe una vista adicional que cae dentro de esta categoría, la cual será accesible por la persona que cuente con el rol de *Revisor* del sistema. El *Revisor* es la persona de Recursos Humanos encargada de atender las inquietudes y comentarios de los empleados en torno a su Información de Nómina. Ordenadas por periodo, la vista mostrará el historial de las solicitudes de aclaración hechas por los empleados asociadas con el documento de Información de Nómina (véase Figura 7.11).

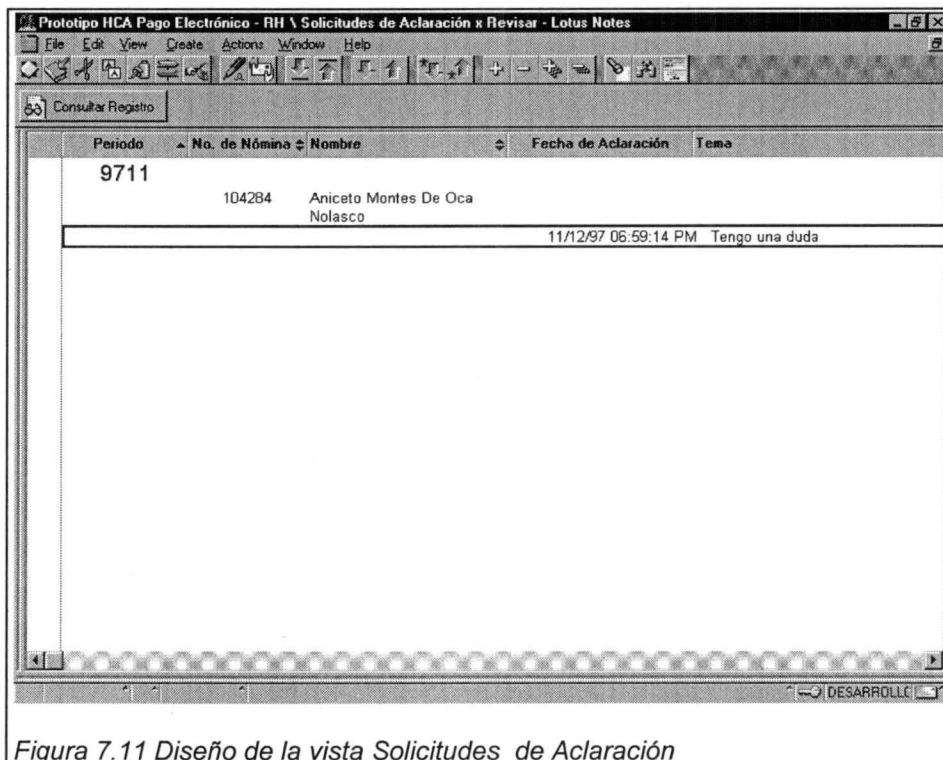


Figura 7.11 Diseño de la vista *Solicitudes de Aclaración*

7.8 *Diseño del Recibo*

En el Anexo I se presenta el diseño del formato del Recibo.

8. CONSTRUCCIÓN DEL SISTEMA

El presente capítulo contiene la información generada durante la Construcción del sistema. Aspectos como la Arquitectura del Sistema y la instalación y configuración de los manejadores de ODBC están contemplados en este capítulo.

8.1 Arquitectura General del Sistema

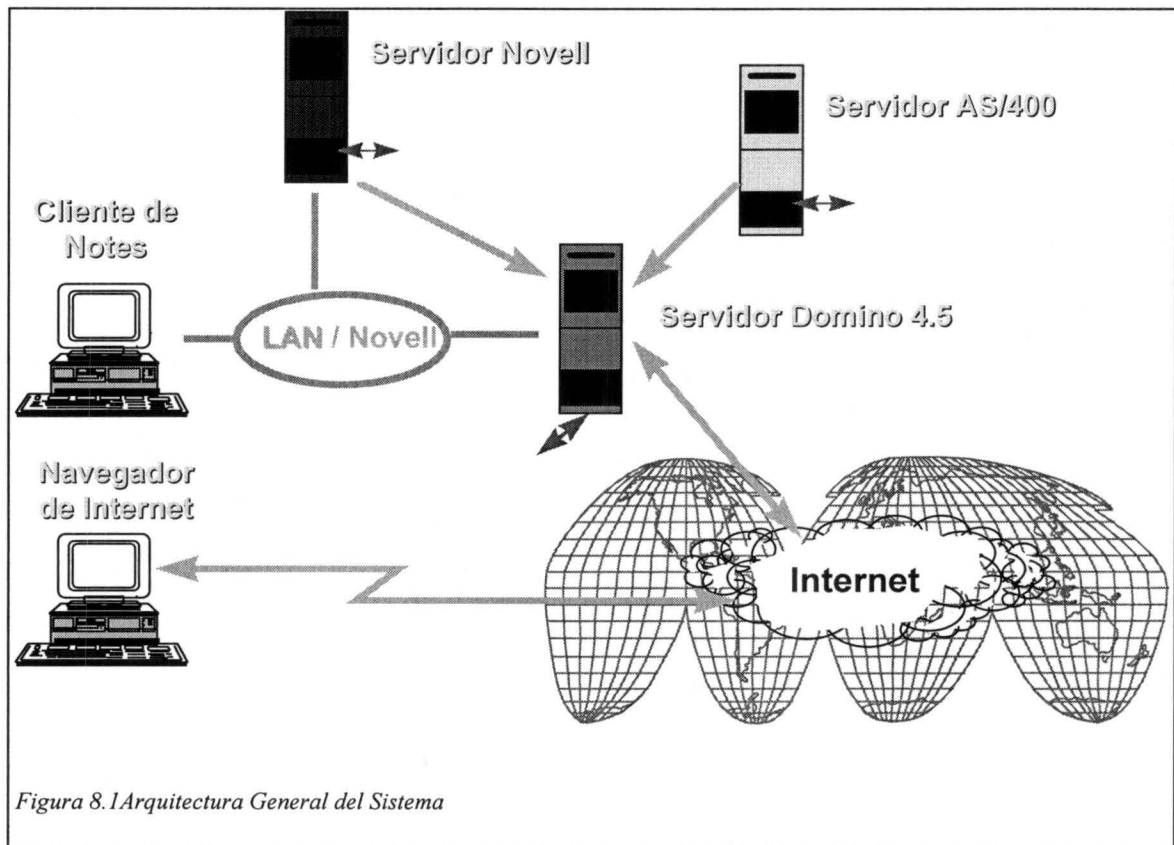
La arquitectura del sistema es la estructura mediante la cual los procesos se comunican. Haciendo una analogía con el cuerpo humano, la arquitectura del sistema es el equivalente al sistema nervioso; el cerebro emite impulsos eléctricos que son transmitidos a través del cuerpo para transmitir sensaciones. De igual forma cualquier estímulo externo es registrado por el cerebro valiéndose de la estructura nerviosa.

Lo mismo sucede con la arquitectura del sistema, existen canales destinados para el flujo de la información entre la fuente del procesamiento de la información y el destino de la misma, el cual puede ser otro dispositivo o el usuario mismo.

Desde un punto de vista simple y general, la arquitectura del sistema de verificación de pago de nómina cuenta con los siguientes elementos básicos:

- Servidor IBM PC Server 320 (Lotus Notes 4.5)
- Servidor IBM AS/400 (donde se ejecuta la Nómina)
- Servidor Compaq Proline 5000 (Servidor de Novell)
- Estación de trabajo Cliente Lotus Notes
- PC con Browser de Internet

Estos elementos se comunican entre sí a través de una INTRANET, la cual está constituida por una LAN y un Fire Wall para el acceso a Internet, utilizando un protocolo de comunicación TCP/IP. La Figura 8.1 ilustra lo mencionado.



8.2 Arquitectura bajo el concepto Cliente / Servidor

Continuando con la analogía del cuerpo humano, el cerebro coordina las acciones ejecutadas por los diferentes miembros del cuerpo. El cerebro emite una orden específica al miembro y éste la ejecuta, entonces podría decirse que el miembro está proporcionando un servicio al cerebro. En este caso podría

decirse que el cerebro está actuando como cliente y el miembro del cuerpo como un servidor. Analizando la arquitectura de un sistema es necesario saber identificar cuando un elemento está proporcionando un servicio a otro, liberando de carga al primero y de tráfico a la red.

Se debe tener cuidado al ubicar estos elementos, ya que, no por el hecho de que un elemento haga el trabajo del otro se convierte esta relación en un concepto cliente / servidor. Al igual que el cerebro tiene sus funciones propias que ningún otro órgano del cuerpo realiza, el cliente deberá tener funciones propias que colaboren con el objetivo del sistema.

Bajo la sombra de lo expuesto, surgen dos puntos de vista. Uno en el que los procesos ejecutados en la estación del usuario funcionan como cliente, y otro en el que los procesos mismos del servidor fungen como cliente del servidor depositador de la información de nómina.

El primer punto de vista se apoya en la naturaleza inherente del lenguaje y la arquitectura misma de la plataforma en que se está desarrollando. A partir de la introducción de Lotus Script en la versión 4.0, Lotus Notes permite hacer referencia a clases front-end y de back-end en la programación de las aplicaciones (ver Anexo II). Dependerá entonces de la habilidad del programador y del tipo de necesidades que resolverá la aplicación la cantidad de trabajo empleado para combinar ambas clases. Aunque no se pretende hacer mucho énfasis en este punto de vista, la Figura 8.2 muestra ejemplo de procesos tipo en los que intervienen procesos front / back end.

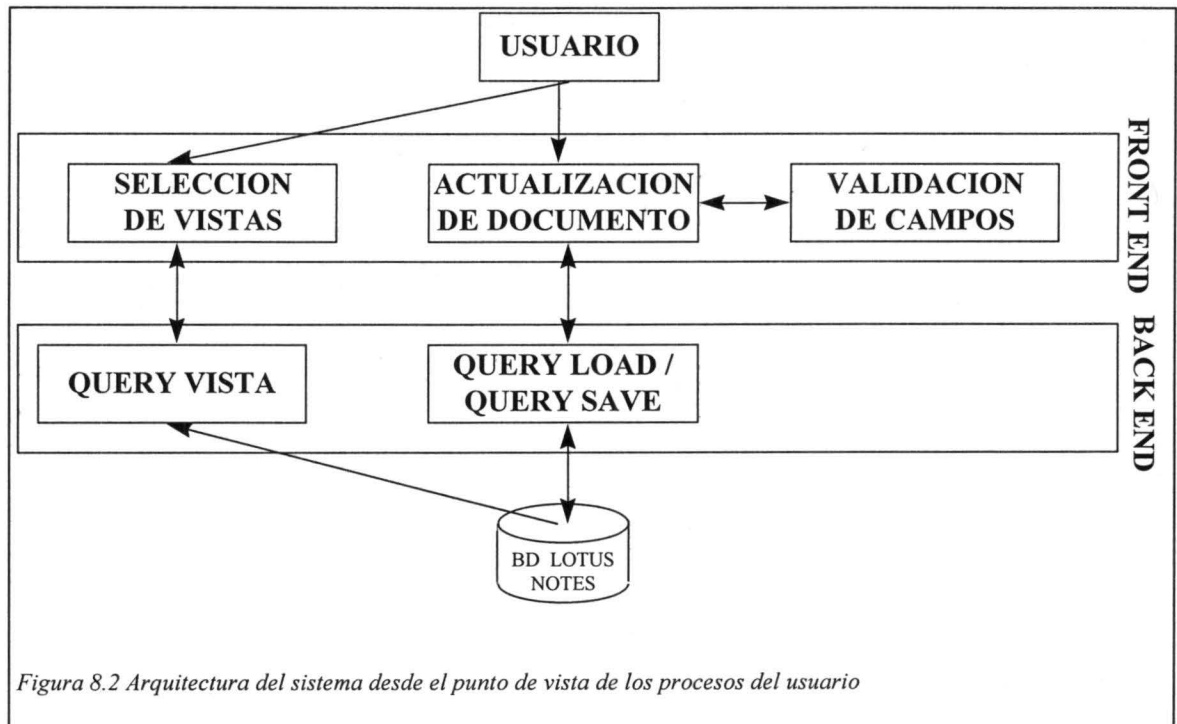
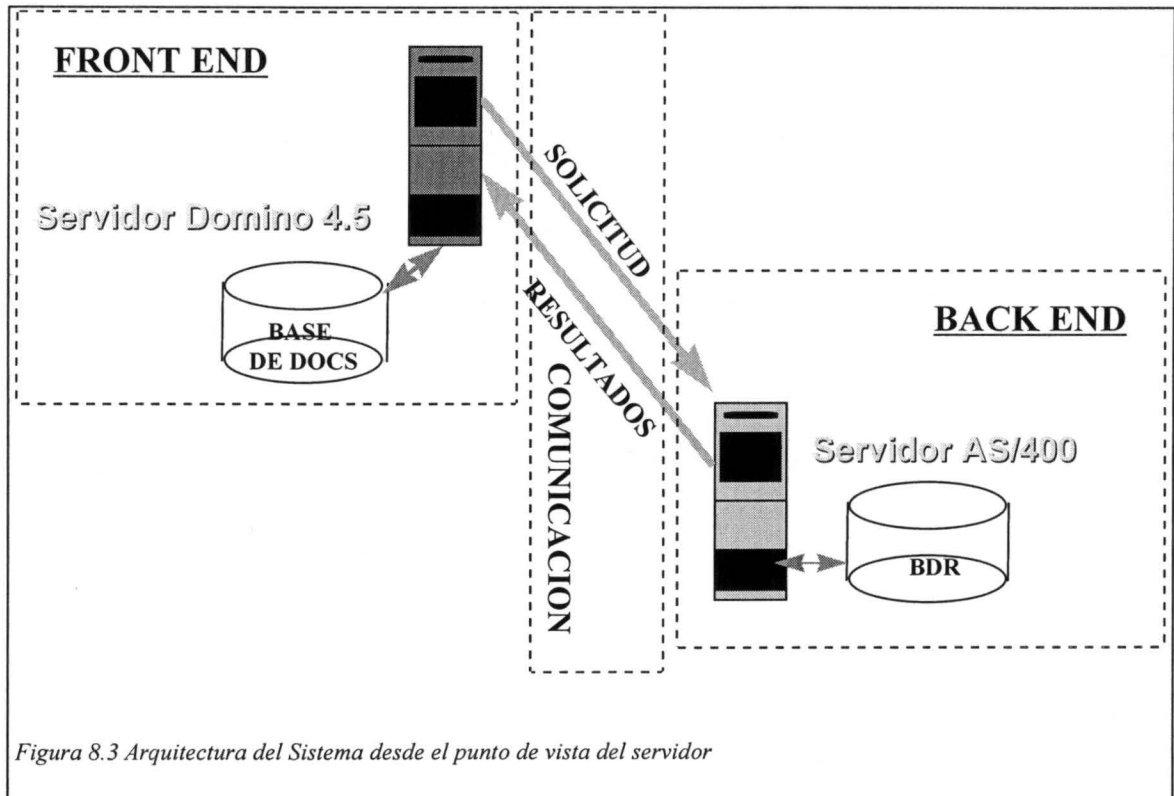


Figura 8.2 Arquitectura del sistema desde el punto de vista de los procesos del usuario

El segundo punto de vista puede parecer un poco confuso en primera instancia, ya que tanto el *cliente* como el *servidor* son ambos servidores. Es importante que cuando hablamos de "*cliente / servidor*" estamos hablando de un **concepto**, el cual es aplicable a cualquier elemento que cumpla con las características propias del concepto.

La aplicación en el servidor de Lotus Notes se vale de las facilidades que le proporciona el manejador de ODBC para enviar una solicitud de información al IBM AS/400, el cual la procesa y regresa **únicamente los resultados**. Una vez que cuenta con ellos, la aplicación se encargará de utilizarlos como mejor le convenga. Como puede apreciarse en la narrativa, la aplicación en el servidor de Lotus Notes (Domino) juega el papel de *cliente* en este caso, y el servidor AS/400 se encarga de proveer los servicios de información que el cliente requiere. La Figura 8.3 ilustra lo anterior.



8.3 Instalación y Configuración del manejador RUMBA

Como se discutió en la sección 5.6, en el ámbito de la Tecnología de Información cada proveedor provee sus propios estándares para la interfaz de sus productos con el exterior. La tecnología ODBC es un esfuerzo para establecer un estándar en la Conectividad de Bases de Datos Abiertas. Existen manejadores particulares para cada base de datos tipo que se encargan de traducir los estatutos de SQL en acciones de consulta específicas.

Existen varios proveedores de manejadores. El manejador ODBC de Rumba²⁴ provee un acceso directo transparente a Bases de Datos Relacionales con Arquitectura Distribuida (DRDA, de sus siglas en inglés), las cuales incluyen DB2[®], DB2/VM, **DB2/400** (para IBM AS/400), y DB2/6000, de esta manera se libera de sobrecarga a los *gateways* de una LAN y a los componentes del servidor.

²⁴ Rumba es una marca registrada de Wall Data Incorporated.

El manejador RUMBA ODBC está contenido dentro del *RUMBA Data Access*, y puede ser utilizado por cualquier aplicación de Windows 95 como Excel® o cualquier otra aplicación desarrollada en dicha plataforma.

8.3.1 Consideraciones antes de la configuración

Antes de instalar y configurar el manejador RUMBA ODBC existen algunos conceptos que se deben considerar; estos conceptos son ciertas tecnologías clave y estándares que están incorporados en el RUMBA Data Access y son:

- Arquitectura de Bases de Datos Relacionales Distribuidas (DRDA²⁵), que consiste en un protocolo de conexión para el procesamiento de bases de datos relacionales (BDR) distribuidas que utilizan las BDR's de los productos de IBM.
- Conectividad de Bases de Datos Abiertas (ODBC²⁶), la cual es una Interfaz de programación para aplicaciones (API²⁷) que permite a los desarrolladores acceder información en un rango amplio de bases de datos, utilizando un simple y consistente conjunto de llamadas a funciones.
- Comunicaciones Avanzadas "Programa-a-Programa" (APPC²⁸), que es una implementación de un protocolo estándar conocido como SNA²⁹, el cual permite interconectar sistemas para comunicar y compartir el procesamiento de los programas.

Para mayor información sobre DRDA y ODBC véase el Anexo II.

²⁵ DRDA, Distributed Relational Database Architecture

²⁶ ODBC, Open Database Connectivity

²⁷ API, Application Programming Interface

²⁸ APPC, Advanced Program-to-Program Communications

²⁹ SNA, Systems Network Architecture

8.3.2 Instalación del manejador

La instalación del manejador RUMBA ODBC se realiza en el servidor de Notes y se realiza en dos pasos.

El primer paso consiste en cerciorarse que tanto el AS/400 como el servidor de Notes cuentan con las librerías básicas DLL que proveen los servicios fundamentales de bases de datos para todos los manejadores en general. Estas librerías normalmente vienen con el sistema operativo del AS/400 y el Windows 95.

En seguida se procederá a instalar el software específico de RUMBA. El RUMBA OFFICE 95/NT es una herramienta de comunicación muy poderosa. Ejecutando el comando Setup.exe del CD de RUMBA OFFICE 95/NT habrá que instalar los elementos llamados *Rumba Data Access* y *AS/400*. Ambos requerirán un espacio total de 50 Mb disponibles.

8.3.3 Configuración del manejador

Para que pueda establecerse una comunicación, el manejador requiere que en el AS/400 esté definida una Base de Datos Relacional (BDR) y que esté configurado un enlace; la forma de verificar el nombre de la BDR es tecleando **WRKRDBDIRE** en la línea de comando del Display, el nombre que aparece en la columna denominada *Relational Database* es el nombre de la BDR.

Si no ha sido definida una BDR en el AS/400 se puede agregar una con el comando **ADDRDBDIRE**. La localidad remota que se solicita debe ser *LOCAL.

Para configurar el enlace existen dos opciones:

1. Añadir un nuevo enlace
2. Configurar un enlace existente

Para añadir un nuevo enlace se debe accesar el icono 32bit ODBC situado en el Panel de Control. Dos tipos de enlace pueden ser agregados. El primero es utilizado exclusivamente por el usuario de la máquina en la que se está agregando el enlace y es llamado *User DSN (User Data Source Names)*. El segundo es similar al anterior, con la diferencia de que el enlace puede ser

utilizado por todos los usuarios que puedan acceder la máquina donde se configuró el enlace; este tipo de enlace es llamado *System DSN (System Data Source Names)*. Es preciso conocer el tipo de enlace requerido antes de comenzar la configuración.

Una vez elegido el tipo de enlace se procede a agregar un nombre de enlace específico; esto se logra presionando el botón de añadir y eligiendo el enlace RUMBA DRDA-32. De aquí en adelante el procedimiento para configurar un enlace nuevo o uno existente es el mismo.

Para configurar el enlace es necesario elegirlo y presionar el botón de configurar. Aparece entonces una caja de diálogo en donde se deberá precisar el nombre del enlace (el cual si ya existe ya está especificado), el nombre de la BDR en el AS/400 y el nombre de la biblioteca en donde se encuentran los datos con los que se trabajará en el AS/400.

Otros datos importantes de configuración son los referentes a la Comunicación. Presionando el botón *Comunicaciones* es posible elegir el tipo de conexión de una lista de proveedores; elegir RUMBA APPC.

Lo siguiente a especificar es el nombre del Modo, de la Unidad Remota (UR) y de la Unidad Local (UL).

El Nombre del Modo determina el número máximo de sesiones de APPC que están disponibles para una conexión. El QPCSUPP es un modo disponible para cualquier AS/400, ya que lo provee IBM, sin embargo para cantidades grandes de datos se recomienda el modo #BATCH.

El nombre de la UR se refiere al identificador del sistema con el que se establecerá la conexión, en este caso el AS/400. Para conocer el nombre de la UR es preciso teclear **DSPNETA** y el nombre del *Local control point name* es el identificador que se busca.

El nombre de la UL es el nombre del servidor desde el cual se establecerá la conexión.

9. Administración y Operación del Sistema

9.1 Documentos para la Administración del Sistema

Una parte medular para el correcto funcionamiento del sistema es constituida por los documentos de *Administración*. El Administrador del Sistema es el responsable de que la información en estos documentos sea la correcta y oportuna, entendiéndose por correcta y oportuna que constantemente refleje la realidad.

Existen dos tipos de documentos de Administración:

1. **Registro de Personal**, el cual se actualiza a través de la vista *Catálogo de Personal* en la categoría *Administración*. Siempre que un empleado nuevo esté registrado en la Nómina, un *Registro de Personal* nuevo deberá ser generado para él especificando su Número de Nómina y su Nombre Notes, en otras palabras, debe haber un *Registro de Personal* para cada Número de Nómina en el Sistema de Nómina en cualquier momento. La Figura 9.1 presenta un ejemplo tipo de un Registro de Personal.

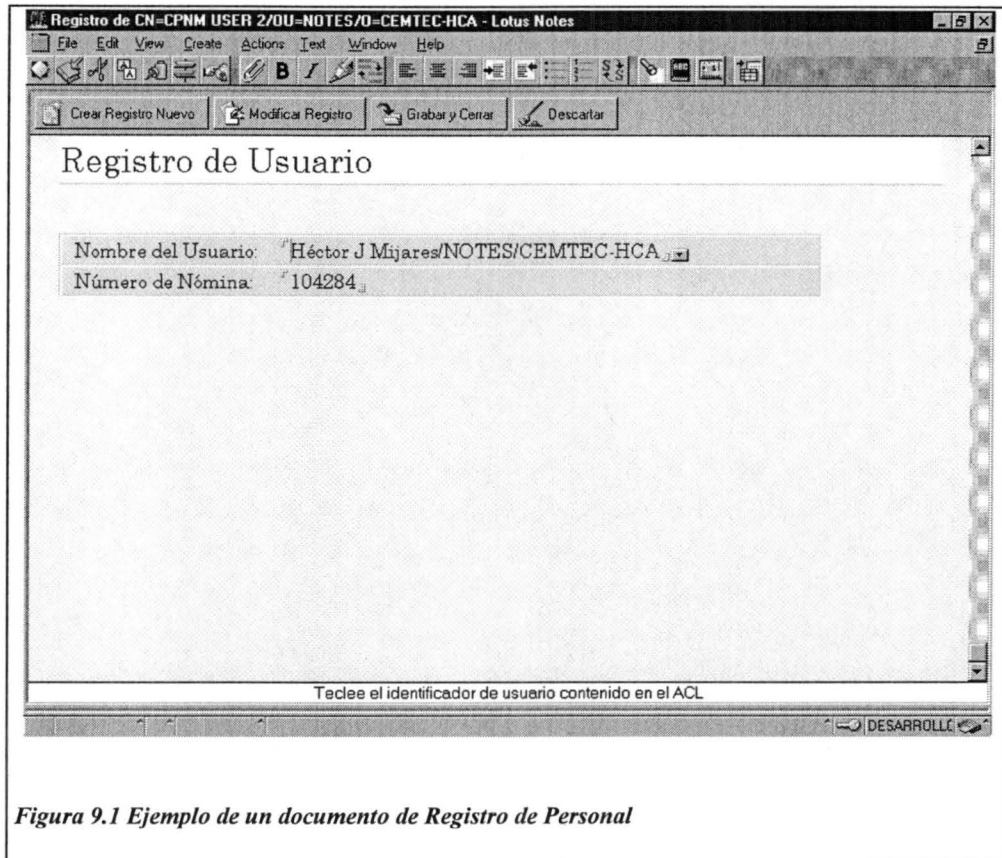


Figura 9.1 Ejemplo de un documento de Registro de Personal

2. **Parametrización**, el cual se actualiza a través de la vista *Configuración* en la categoría *Administración*. Siempre debe existir sólo un documento de *Parametrización* y nunca debe eliminarse. Este documento se ilustra en la Figura 9.2 y contiene la información exacta sobre la Fuente de la información de Nómina, rutas de acceso, nombre de los archivos fuente y mensajes a utilizar en los procesos de notificación y recordatorio. Es de vital importancia que los datos aquí contenidos sean los correctos, ya que el proceso de importación de nómina depende de ello para tener éxito.

9.2 Roles y niveles de acceso en la Administración del Sistema

El sistema contempla 5 roles para su funcionamiento:

1. **[Empleado]**. Todo usuario del sistema cuya información de nómina esté disponible en las tablas que arroja el proceso de Nómina deberá tener asignado este rol. El [Empleado] puede consultar y firmar exclusivamente los documentos que contienen su propia información, y a partir de éstos puede

ligar Solicitudes de Aclaración para enviarlas al encargado por parte de Recursos Humanos.

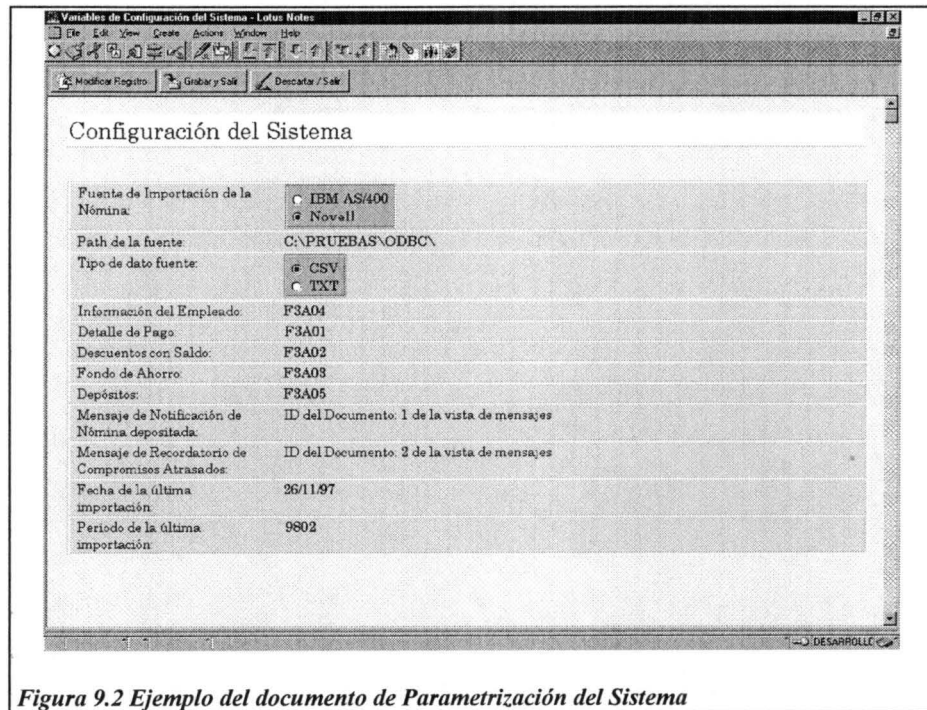


Figura 9.2 Ejemplo del documento de Parametrización del Sistema

2. **[RH]** (Recursos Humanos). El [RH] es el usuario cuyos privilegios le permiten revisar los documentos de todos los empleados a través de las vistas que fueron designadas para ello. El [RH] además puede activar en forma manual el proceso de Importación de la Nómina, el proceso de Notificación a Usuarios y el proceso de envío de Recordatorios.
3. **[Revisor]**. El [Revisor] es un complemento del rol [RH]. Un [Revisor] es el encargado de atender las inquietudes y sugerencias manifestadas en las solicitudes de aclaración. El sistema detecta automáticamente el nombre Notes de el (los) revisor(es) para ubicarlos como destinatarios de correo para el envío de la aclaración.
4. **[Observador]**. Al igual que el [RH], el [Observador] puede revisar los documentos de información de nómina de todos los empleados, con la diferencia de que el [Observador] no tiene los privilegios necesarios para activar los procesos arriba citados. La idea de la existencia de este rol se debe a la necesidad de que algunos gerentes puedan revisar los sueldos de los empleados.

5. **[Administración]**. El usuario *administrador* del sistema (el cual posee este rol) deberá vigilar el correcto funcionamiento de la herramienta parametrizando el documento de *Parametrización del Sistema*. Será responsabilidad de éste también el mantener constantemente actualizado el *Catálogo de Personal* y la lista de control de accesos (ACL) a la Herramienta.

En cuanto a los niveles de acceso se refiere, cada usuario del sistema que posea los roles de [Empleado], [RH], [Observador] o [Revisor] deberán tener asignado el nivel de acceso **Author** en la lista de control de accesos (ACL) y el usuario con el rol de [Administrador] deberá de asignársele el nivel de acceso **Editor**.

9.3 Aspectos de la Operación del Sistema

La operación del sistema se encuentra distribuida en 5 procesos:

1. Importación de la Nómina
2. Notificación al Usuario
3. Envío de Recordatorios
4. Revisión del Pago de Nómina
5. Revisión de Solicitud de Aclaración

Los primeros tres preferentemente se realizarán en forma automática a través de los agentes programados a través de un calendario según las políticas de la empresa en cuanto a estos procesos se refiere, sin embargo los procesos pueden ser activados en un momento dado por los usuarios con el rol [RH]. Estos procesos están ligados fuertemente a la información contenida en el documento de *Parametrización del Sistema*.

El proceso de Importación de la Nómina requiere además que los manejadores de ODBC se encuentren instalados y correctamente configurados en el servidor donde radica la aplicación.

El proceso de Revisión del Pago de Nómina consiste en el acceso al documento de Información de Nómina; el dueño del documento, el empleado cuyo número de nómina corresponde al del documento, puede firmarlo electrónicamente, imprimirlo o bien construir y enviar una solicitud de aclaración al revisor de

Recursos Humanos. Para los procesos de Notificación al Usuario y Envío de Recordatorios es necesario tener definidos mensajes diseñados para este propósito. La Figura 9.3 muestra un ejemplo de un documento *Mensaje de Notificación*; todo mensaje deberá contar con un identificador numérico único, el cual será relacionado con el proceso adecuado en el documento de *Parametrización del Sistema*.

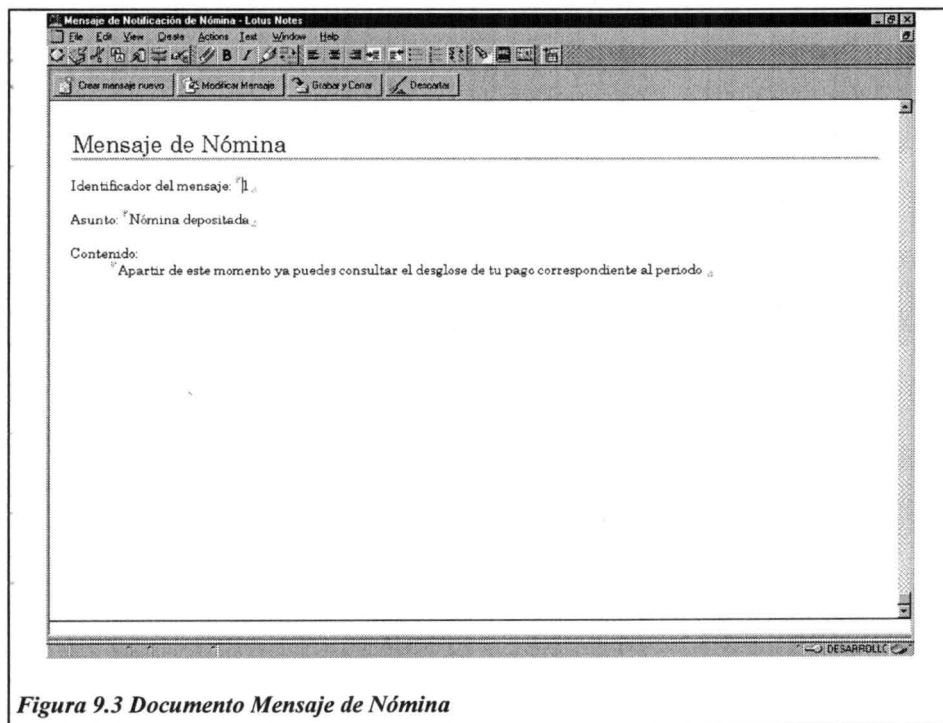


Figura 9.3 Documento Mensaje de Nómina

En el proceso de Revisión de Solicitud de Aclaración interviene el usuario con el rol [Revisor] para dar lectura y respuesta a las inquietudes de los empleados. La fds muestra un ejemplo de una Solicitud de Aclaración.

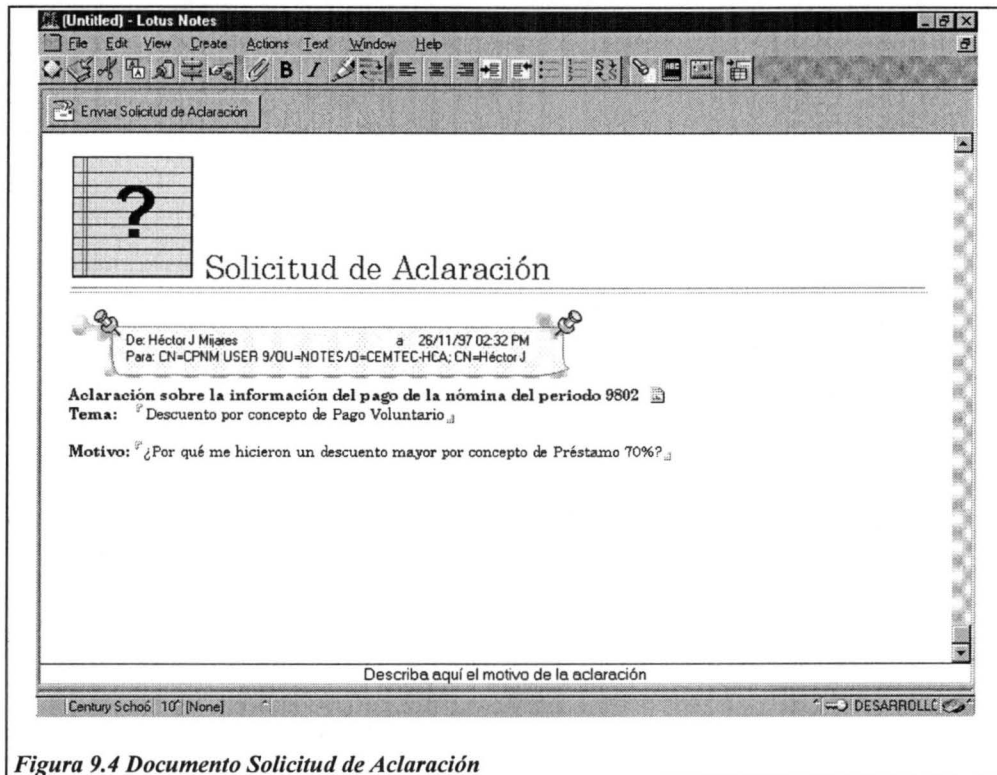


Figura 9.4 Documento Solicitud de Aclaración

9.4 Delimitación de Responsabilidades

Como ya se ha expuesto, la información del documento de Parametrización del Sistema es crítica para el correcto funcionamiento de la herramienta y es responsabilidad del administrador mantenerla actualizada.

En lo tocante a la Seguridad de la Información, es responsabilidad del Administrador del Sistema del correcto respaldo periódico de la información de la herramienta. Preferentemente el respaldo se realizará por lo menos una vez después del proceso de importación de la nómina.

Es responsabilidad de Cemtec el instalar el producto adecuadamente para su funcionamiento, incluyendo la configuración (y no adquisición) de el (los) manejador(es) de ODBC requeridos por la herramienta.

Aspectos referentes a la comunicación entre el servidor Windows NT donde reside la aplicación y el AS/400 o el servidor de archivos Novell que vayan más allá de la configuración del manejador de ODBC están fuera del alcance de esta herramienta. Así también, es responsabilidad de la empresa cliente el contar con

los archivos necesarios para el manejo de ODBC y con una infraestructura de red que soporte las características descritas en relación al acceso desde un *browser*.

10. CONCLUSIONES

10.1 Metodologías de desarrollo

Aunque la Metodología Orientada a Objetos es ampliamente utilizada y promete mucho desde el punto de vista "natural" en que se realiza el Análisis y el Diseño Orientado a Objetos, en lo que respecta a un Workflow aún falta mucho por hacer. Es necesario establecer modelos "tipo" cuyo enfoque este centrado en este tipo de concepto, ampliar nuestra visión para no pensar exclusivamente en términos de "procesos" o "datos" como lo ha sido hasta hoy, sino incorporar además objetos que incluyan "reglas del negocio" o mejor aún incluir *roles* que representen la función que tienen las personas como sujeto dentro del sistema, ya que, a fin de cuentas un sistema aislado no tiene ninguna utilidad.

El Análisis y el Diseño de este proyecto se ha desarrollado bajo la perspectiva muy particular de James Martin y se ha encontrado que la línea que él plantea como guía para el análisis y el diseño es muy vaga y poco comprometida para sistemas que se salen de lo tradicional, por lo tanto para el desarrollo de futuros proyectos de naturaleza similar al presente se recomienda considerar las propuestas de otros autores como Grady Booch para etapas tempranas del

desarrollo, o bien considerar el toque personal que se le ha dado a esta metodología como adaptación a las necesidades de este proyecto.

10.2 Tecnologías empleadas

Se ha dicho que la tecnología Cliente - Servidor distribuye las cargas de procesamiento y reduce el tráfico en la red; Lotus Notes 4.5 a través de Lotus Script provee un conjunto de clases para front y back end que pueden ser manipuladas por separado dejando las tareas de validación y selección de información al front end y las tareas de almacenamiento y búsqueda al back end, sin embargo es importante hacer notar dependerá de la naturaleza de la aplicación el que se explote adecuadamente esta tecnología, ya que algunas aplicaciones tal vez se centrarán exclusivamente en una funcionalidad deseada sin tomar en cuenta la carga de procesamiento en el cliente y el servidor.

El empleo de la tecnología de Conectividad de Bases Abiertas (ODBC) también se vale de los beneficios de una arquitectura cliente servidor para realizar su tarea. A pesar de que esta tecnología es una muy buena aproximación para el establecimiento de un estándar de conectividad entre bases de datos distintas, aún existen áreas de oportunidad en cuanto a la configuración de los manejadores se refiere. La ventaja más grande que existe de emplear la tecnología de ODBC es que basta con establecer un conjunto de consultas (query's) utilizando el estándar SQL para que a través de manejadores de ODBC correctamente configurados pueda establecerse conectividad en más de una base de datos externa a la aplicación. Inherente a esta tecnología está también la tecnología Cliente / Servidor; en el caso particular de este proyecto la HCA hace las veces del cliente al hacer una solicitud de búsqueda de información hacia el IBM AS/400, el cual procesa esa solicitud interpretándola a través del manejador de ODBC y convirtiéndose así en el servidor, en este esquema. Algo similar ocurre en todos los manejadores de ODBC, proveyendo a las aplicaciones de mucho poder.

10.3 Plataforma de desarrollo

Con la introducción de Lotus Script en la versión 4.0 se ha incrementado la flexibilidad en el desarrollo de aplicaciones con respecto a las versiones anteriores, sin embargo aún ahora en la versión 4.5 Lotus Script posee deficiencias considerables que limitan al desarrollador en la construcción de aplicaciones. Lotus Notes por su naturaleza es ideal para el Desarrollo Rápido

de Aplicaciones o alguna otra técnica de prototipo, ya que desde etapas tempranas del diseño es posible integrar al usuario en la construcción de la herramienta y ayudarlo a definir claramente los requerimientos.

Domino Server es una buena alternativa para la comunicación remota a un bajo costo, aprovechando la infraestructura de la red mundial Internet, pero a pesar de las grandes ventajas de publicación de bases de datos que ofrece la transición aún no es tan transparente como se presume para el desarrollador. Es conveniente estudiar las ventajas que promete la versión 4.6 con respecto a la anterior, al parecer tiene incorporadas algunas mejoras en cuanto al manejo de las formas desde el acceso de un browser.

10.4 Conclusiones generales

El software de Trabajo en Grupo es el más indicado para resolver problemas de tipo Flujo de Trabajo, sin embargo, en el desarrollo de herramientas bajo este concepto es necesario romper con los paradigmas rígidos que hasta ahora se han venido manejando en el desarrollo Tradicional y en el Orientado a Objetos, de manera que se pueda obtener un producto de alta calidad cuyas especificaciones funcionales sean lo suficientemente claras para que la transición entre el análisis y el diseño sea más natural.

Aunque Lotus Script es un lenguaje orientado a eventos, no es posible tratarlo de la manera tradicional en la que el software está dividido en "módulos". Todos sus aspectos procedurales están repartidos en los diferentes elementos del diseño, lo cual dificulta la programación secuencial en el sentido estricto en el que se está acostumbrado, ya que es necesario considerar el estado de los diferentes elementos del diseño para definir una condición para una acción.

11. BIBLIOGRAFÍA

11.1 BIBLIOGRAFÍA DE REFERENCIAS

- [CAPIONE, 96] Capione Mary
THE JAVA: TUTORIAL; OBJECTS ORIENTED
PROGRAMMING FOR INTERNET
Ed Addison/Wesley Publishing
EUA 1996
- [GRAHAM, 95] Graham, Ian S.
HTML: Source Book
Ed. Wiley
EUA 1995
- [LOTUS,97] Lotus Development Corporation
LOTUS NOTES
APLICACION DEVELOPMENT REFERENCE
Lotus
EUA, 1997
- [LOTUS, 96a] Lotus Development Corporation.
White Paper: Lotus and the Internet
EUA, 1996
- [LOTUS, 96b] Lotus Development Corporation.
Lotus Internotes 1.0
EUA, 1996
- [LOTUS, 96c] Lotus Development Corporation.
Creating Web Applications with Domino
EUA, 1996
- [MARTIN,94] Martin James, Odell James J.
ANALISIS Y DISEÑO ORIENTADO A OBJETOS
Prentice Hall
México, 1994

- [POMPILI,95] Pompili Tony
LET'S TALK (DISCUSION ACERCA DE SOFTWARE DE
CORREO ELECTRONICO) en : PC-Magazine
EUA, Noviembre 21, 1995; v14, n20 pp 205-234
- [SENN,92] Senn, James A.
ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN
Mc Graw Hill, 2a. Edición
México, 1992
- [WINDOWS95,95] Windows 95 Help Context
“Connectivity”, “ODBC Help”
Microsoft

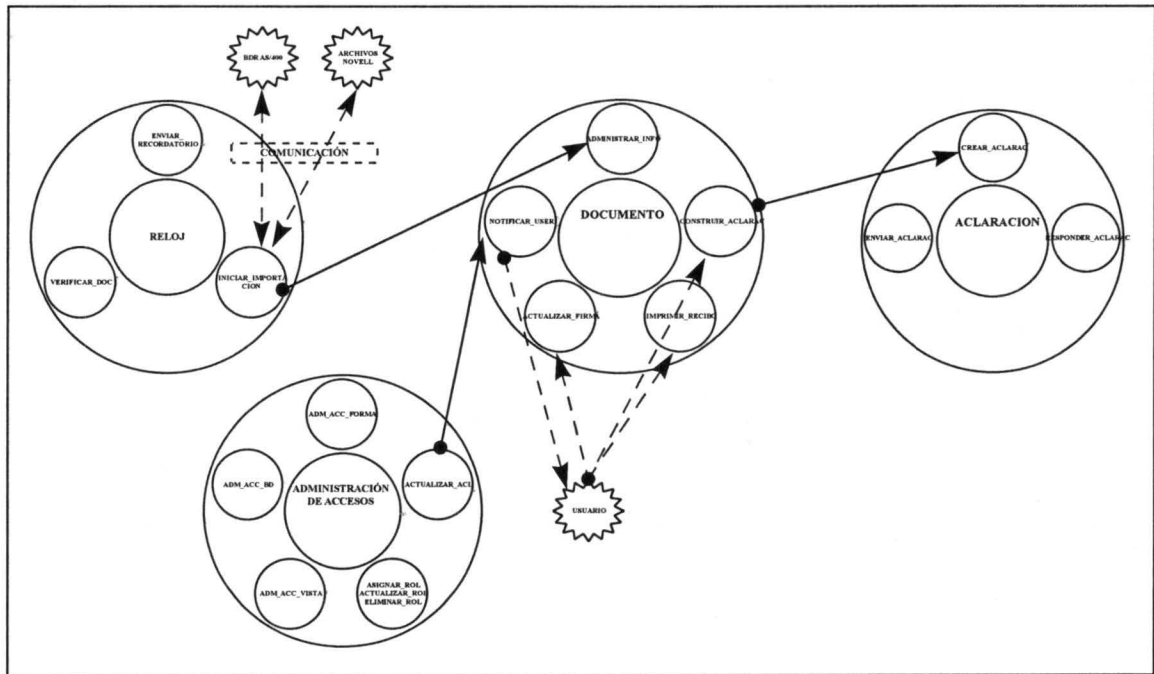
11.2 BIBLIOGRAFÍA GENERAL

- [COLEMAN, 95] Coleman, David E.
GROUPWARE '95; The Groupware Solutions Conference
Ed. Omnipress
EUA, Agosto, 1995
- [DATE,90] Date C. J.
AN INTRODUCTION TO DATABASE SYSTEM
Addison Wesley
EUA, 1990
- [JENNINGS, 94] Jennings, Roger
Database DEVELOPER'S GUIDE with Visual Basic 3
SAMS PUBLISHING
EUA, 1994
- [LOTUS,96d] Lotus Development Corporation.
Developing Applications with Lotus Notes R.4.
EUA, 1996
- [LOTUS, 96e] Lotus Development Corporation.
White Paper: Lotus Notes

- EUA, 1996
- [LOTUS, 96f] Lotus Development Corporation.
Domino and InterNotes
EUA, 1996
- [LOTUS, 96g] Lotus Development Corporation.
Programmer's Guide
EUA, 1996
- [LOTUS, 96h] Lotus Development Corporation.
Lotus Script release 2.0
EUA, 1996
- [LOTUS, 96i] Lotus Development Corporation.
Application Developer's Guide
EUA, 1996
- [PRESSMAN,93] Pressman, Roger S.
INGENIERÍA DEL SOFTWARE: UN ENFOQUE PRACTICO
Mc Graw Hill, 3era. Edición
España, 1993
- [ZIMMERMAN, 96] Zimmerman, Scott
BUILDING AN INTRANET WITH WINDOWS NT
Macmillan Computer Publishing
EUA. 1996

APÉNDICE A

Desde que se está diseñando un *Workflow*³⁰, el sistema tendrá que estar interactuando constantemente con entidades externas que bajo nuestro enfoque son también tipos de objetos.



Escenario VI Interacción de objetos con el exterior a través de sus métodos

Por ejemplo, en el Escenario VI el límite del sistema a través del método NOTIFICAR_USER de la clase DOCUMENTO es rebasado en el momento en que el control se transfiere al sistema de correo electrónico de Lotus Notes. A partir de este momento el sistema “ya no tiene nada que hacer”, sino esperar un estímulo del ambiente (una entidad externa) que active uno o más métodos para continuar con su trabajo. Este estímulo puede presentarse como un evento externo, o bien como una orden directa hacia un método del sistema. Un ejemplo de esto último podría ser cuando el usuario “decide” firmar el documento, o imprimir la información, o bien solicitar una aclaración.

³⁰ Véase la sección LOTUS NOTES

Es importante recalcar que nuestro análisis no se está basando en *decisiones* sino en *activaciones*. No basta con que el usuario “decida” firmar, tendrá que activar el método apropiado para que la afectación del sistema tenga efecto.

APÉNDICE B

Diseño de la Base de Documentos

Forma:	Información de Nómina		
Campo	Tipo de Dato	Tipo de Campo	Descripción
Z9NPRD	Número	Calculado al crear	Número de Periodo que se paga
Z9FPRD	Tiempo	Calculado al crear	Fecha del Recibo
Z4\$SMN	Texto	Calculado al crear	Sueldo Mensual
Z1PHRW	Número	Calculado al crear	Días trabajados
Z4AN8	Número	Calculado al crear	Número de Nómina
Nombre_Completo	Texto	Calculado al crear	Nombre y Apellidos del empleado
Z4\$RFC	Texto	Calculado al crear	Registro Federal de Causantes
Z4\$IMS	Texto	Calculado al crear	No. de registro de Seguro Social
Z4\$CBC	Texto	Calculado al crear	No. de cuenta bancaria
IstPDBA_P	Texto Multivalor	Calculado al crear	Conceptos de percepción
IstGPA_P	Currency Multivalor	Calculado al crear	Importe de percepción
totGPA_P	Currency	Calculado al crear	Total de percepciones
IstPDBA_D	Texto Multivalor	Calculado al crear	Conceptos de deducción
IstGPA_D	Currency Multivalor	Calculado al crear	Importe de deducción
totGPA_D	Currency	Calculado al crear	Total de deducciones
IstPDBA_PV	Texto Multivalor	Calculado al crear	Conceptos de Pagos Voluntarios
IstGPA_PV	Currency Multivalor	Calculado al crear	Importe de Pagos Voluntarios
totGPA_PV	Currency	Calculado al crear	Total de Pagos Voluntarios
IstFOA	Currency Multivalor	Calculado al crear	Aportaciones para el Fondo de Ahorro
NETO	Currency	Calculado al crear	Cantidad Neta a Recibir
FIRMA	Especial	Calculado al	Indicador de que el

APÉNDICE B

	Check Box	crear	documento fue electrónicamente firmado
Empleado_CN	Nombre (escondido)	Calculado al crear	Nombre (Notes) Común del usuario que firma.
dt_Firma	Tiempo (escondido)	Calculado al crear	Fecha de la firma del documento
Control	Texto (escondido)	Calculado al crear	Indicador del estado del documento
Lectores	Readers (escondido)	Calculado al crear	Lista de quienes pueden leer el documento
Autor	Author (escondido)	Calculado al crear	Nombre Notes del empleado relacionado con el docto.
Z4\$NBR	Texto (escondido)	Calculado al crear	Nombre de pila del empleado
Z4\$APA	Texto (escondido)	Calculado al crear	Apellido Paterno del empleado
Z4\$AMA	Texto (escondido)	Calculado al crear	Apellido Materno del empleado

Forma:		Solicitud de Aclaración	
Campo	Tipo de Dato	Tipo de Campo	Descripción
xZ9NPRD	Número	Calculado al crear	Número de Periodo del cual se solicita aclaración
Referencia	Texto enriquecido	Calculado al crear	Enlace hacia el documento de Nómina
Tema	Texto	Editable	Breve abstracto de la aclaración
Comentario	Texto enriquecido	Editable	Motivo de la aclaración
Remitente	Nombre	Calculado al crear	Nombre Notes de quien envía la solicitud
SendTo	Nombre	Calculado al crear	Nombre(s) del (los) destinatario(s). El Revisor.
Fecha	Fecha	Calculado	Fecha de creación del comentario.

Forma:		Configuración	
Campo	Tipo de Dato	Tipo de Campo	Descripción
fuelle	Especial Radio Button	Editable	Nombre de la fuente de información: AS/400, Novell
biblioteca	Texto	Editable	Nombre de la biblioteca en el IBM AS/400
path	Texto	Editable	Ruta de acceso a través de Novell; v.gr. "F:\ODBC\"

APÉNDICE B

ext	Texto	Editable	Extensión del tipo de dato fuente en Novell; vgr. TXT
idNotificación	Especial Diálogo	Editable	Identificador del mensaje a enviar para la notificación
idRecordatorio	Especial Diálogo	Editable	Identificador del mensaje a enviar como recordatorio
dtImportación	Fecha	Calculado al crear	Fecha de la última importación
nPeriodo	Número	Calculado al crear	Número de periodo de la última importación

Forma:	Mensajes de Nómina		
Campo	Tipo de Dato	Tipo de Campo	Descripción
MsgID	Texto	Editable	Identificador único para un mensaje
Tema	Texto	Editable	Abstracto representativo del mensaje
MSG	Texto	Editable	Contenido del mensaje

Forma:	Registro del Usuario		
Campo	Tipo de Dato	Tipo de Campo	Descripción
catUserName	Nombre	Editable	Nombre Notes del empleado
catNoNomina	Número	Editable	Número de Nómina del empleado

ANEXO I - Recibo de Nómina

Nombre de la Empresa

Recibo de Nómina

PERIODO	FECHA	SUELDO	DIAS TRABAJADOS	RFC	No IMSS	No CUENTA CHEQUES
No EMPLEADO		NOMBRE DEL EMPLEADO				

PERCEPCIONES	IMPORTE	DEDUCCIONES	IMPORTE
CONCEPTO_P 1	\$0,000.00	CONCEPTO_D 1	\$000.00
CONCEPTO_P 2	\$0.00	DEPOSITO EN CUENTA XXXXXX	\$0,000.00
CONCEPTO_P 3	\$0.00		
TOTAL DE PERCEPCIONES	\$0,000.00	TOTAL DE DEDUCCIONES	\$0,000.00
		PAGOS VOLUNTARIOS	IMPORTE
		PRESTAMO	\$000.00
		TOTAL DE PAGOS VOLUNTARIOS	\$000.00

RECIBI NETO:	\$0.00
---------------------	---------------

Por este medio y al declararme satisfecho hago constar que he recibido de la empresa a través de depósito bancario en la(s) cuenta(s) arriba mencionada(s) y a mi entera conformidad, la cantidad especificada en TOTAL DE PERCEPCIONES, la cual corresponde a mi sueldo relativo a las horas y jornadas trabajadas por mí durante el período que hoy termina, incluyendo días de descanso semanales y legales. Estoy de acuerdo en que la empresa haga todas las deducciones que por ley procedan y retenga los pagos a los que voluntariamente me he comprometido y que se muestran desglosadas con sus saldos correspondientes. Con la cantidad que se me entrega, la empresa no me adeuda cantidad alguna por concepto de sueldo, diferencias de salarios, horas extras o cualquier otra percepción a que yo tengo derecho a la fecha del presente recibo

FIRMA

ANEXO II

Conceptos de Configuración de manejadores de ODBC

Arquitectura de Bases de Datos Relacionales Distribuidas (DRDA)

El DRDA es el protocolo de acceso a los datos que utiliza RUMBA para conectarse a las BDR's de los productos de IBM.

Esta arquitectura cada vez es más aceptada como un estándar para conectividad de bases de datos, y es ampliamente soportada por otras compañías que desarrollan Sistemas Administradores de Bases de Datos (DBMS), aplicaciones que desarrollan bases de datos y herramientas relacionadas.

Existen dos elementos principales en la conexión, la aplicación local conocida como *Local Unit (LU)* y la unidad remota de trabajo (RUW) la cual normalmente la constituye un *host*. La LU constituye el front-end y la RU el back-end, ya que estamos hablando de una arquitectura cliente servidor.

La idea general es que una aplicación en un *site* (LU) puede enviar solicitudes a bases de datos de una unidad de trabajo dada (RUW) para su ejecución en un *site* remoto. La LU puede revisar los resultados de las solicitudes antes de procesar sus transacciones. Todo el procesamiento es efectuado en el *site* remoto. Esto provee a los usuarios máxima flexibilidad en la construcción de transacciones en una PC aplicando todo el poder de una base de datos de un *host* al procesamiento actual de transacciones.

Conectividad de Bases de Datos Abiertas (ODBC),

El ODBC es una Interfaz de programación para aplicaciones (API) que permite a los desarrolladores acceder información en un rango amplio de bases de datos, utilizando un simple y consistente conjunto de llamadas a funciones.

Este estándar de Microsoft ofrece a las aplicaciones de Windows una manera única de acceder múltiples bases de datos en forma simultánea utilizando SQL, y administrar los datos obtenidos de las consultas de manera que el usuario final pueda importar fácilmente los datos en sus aplicaciones.

El RUMBA Data Access soporta todas las funciones del nivel 1 y la mayoría del nivel 2 (véase sección TECNOLOGÍA ODBC del capítulo 5), dando a los usuarios la capacidad de manipulación y definición de datos.

Anexo III

Especificaciones técnicas, Formula Language

Database Information:

Server:	CN=HCA- NOTES1/O=CEMTEC-HCA DESCEMEX\PAGO.nsf
Path:	Prototipo HCA Pago Electrónico
Title:	17/10/97 05:40:41 PM
Date Created:	02/12/97 04:57:06 PM
Last Data Modification:	02/12/97 07:50:32 PM
Last Non-Data Modification:	[None Assigned]
Categories:	[Not Assigned]
Template Name:	[Not Assigned]
Inherit Design From Template:	Disabled
Text Compression:	Disabled
Full-Text Indexing:	[Not Assigned]
Local Encryption Key Name:	

Space Usage Information:

On-Disk Size:	2016 Kilobytes
Data Documents:	77
Space Used:	1673 Kilobytes
Space Free:	374 Kilobytes
% Space Free:	18.3%

Replication Information:

Replica ID:	A3256528:0044730A
Priority:	Medium
Propagate Deletions:	Enabled
Replication:	Enabled
Cutoff Document Deletion:	Disabled
Truncate Large Documents:	Disabled
Replicate Title:	Enabled
Replicate Categories:	Enabled
Replicate Design Template Names:	Enabled

Other Information:

List In Database Catalog:	Enabled
List In 'Open Database' Dialog:	Enabled
Hide Database Design:	Disabled
Background Agents:	Enabled

Access List Information:

User/Group Name:	-Default-
Access Level:	No Access
Role(s):	[None Assigned]
User/Group Name:	OtherDomainServers

Access Level:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Delete Documents:
Role(s):

User/Group Name:
Access Level:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):
User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):

Manager
Yes
[None Assigned]
**CN=HCA-
NOTES1/O=CEMTEC-HCA**
Manager
Yes
[Empleado], [RH],
[Observador],
[Administración]
LocalDomainServers
Manager
Yes
[None Assigned]
**CN=CPNM USER
9/OU=NOTES/O=CEMTEC-
HCA**
Author
Yes
No
[Empleado], [RH], [Revisor]
**CN=CPNM USER
8/OU=NOTES/O=CEMTEC-
HCA**
Author
Yes
No
[Empleado], [Observador]
**CN=CPNM USER
7/OU=NOTES/O=CEMTEC-
HCA**
Author
Yes
No
[Empleado], [RH]
**CN=CPNM USER
5/OU=NOTES/O=CEMTEC-
HCA**
Author
Yes
No
[Empleado]
**CN=CPNM USER
4/OU=NOTES/O=CEMTEC-
HCA**
Author
Yes
No
[Empleado]

User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):

CN=CPNM USER
3/OU=NOTES/O=CEMTEC-
HCA
Author
Yes
No
[Empleado]

User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):

CN=CPNM USER
2/OU=NOTES/O=CEMTEC-
HCA
Author
Yes
No
[Empleado]

User/Group Name:

Access Level:
Can Create Documents:
Can Delete Documents:
Role(s):

CN=CPNM USER
10/OU=NOTES/O=CEMTEC
-HCA
Author
Yes
No
[Empleado], [RH],
[Administración]

User/Group Name:

Access Level:
Role(s):

CN=CPNM USER
1/OU=NOTES/O=CEMTEC-
HCA
Reader
[Empleado]

User/Group Name:

Access Level:
Can Delete Documents:
Role(s):

CN=Héctor J
Mijares/OU=NOTES/O=CE
MTEC-HCA
Manager
Yes
[Empleado], [RH],
[Observador], [Revisor],
[Administración]

Shared Field Information:

[No Shared Fields Defined]

Form Information:

Form:

Form Alias:
Last Modification:
Type:
Window Title Formula:

Include in Compose Menu:
Include in Query by Form:
Default Database Form:

Configuración

frmConfiguración
01/12/97 01:21:03 PM
Document
SELECT "Variables de
Configuración del Sistema"
Yes
No
No

Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	[RH] [Revisor]
Field:	fuentes
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Radio Buttons
Allowable Keywords:	IBM AS/400 Novell
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	path
Datatype:	Text
Help Description:	Teclee el path donde se encuentra la información de nómina, v.gr. F:\InfoNomina\
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F:ODBC"
Input Translation Formula:	@UpperCase(path);
Field:	biblioteca
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"RHDATA/"
Field:	ext
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Radio Buttons
Allowable Keywords:	CSV TXT
Sign When Mailing/In Section:	No

Encryption:	Disabled
Update Requires Editor Access:	No
Field:	arcInfoEmpleado
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F3A04"
Field:	arcDetallePago
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F3A01"
Field:	arcDescSaldo
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F3A02"
Field:	arcFondoAhorro
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F3A03"
Field:	arcDepósitos
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"F3A05"
Field:	arcControl
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	msgNotificación

Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	msgRecordatorio
Datatype:	Keywords
Help Description:	[Not Assigned]
Field Type:	Editable
Keyword User Interface:	Standard
Allow Values Not In List:	No
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	dtlImportación
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	nPeriodo
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	0

Form:

Form Alias:
 Last Modification:
 Type:
 Window Title Formula:

 Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:

Información de Nómina

Principal
 02/12/97 07:49:06 PM
 Document
 SELECT "Información de
 Nómina de " +
 @ProperCase(Z4\$NBR + " "
 + Z4\$APA + " " + Z4\$AMA);
 No
 Yes
 Yes
 No

Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Field:	Z4\$NBR
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$APA
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$AMA
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	FULLNBR
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Field:	\$Yo
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled

Update Requires Editor Access:	No
Formula:	@DocumentUniqueID
Field:	Control
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	"Pendiente"
Field:	Lectores
Datatype:	Reader Names
Input Multi-Value Separator(s):	Comma, Semicolon, New Line, Blank Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Empleado
Datatype:	Author Names
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Revisor
Datatype:	Reader Names
Input Multi-Value Separator(s):	Comma, Semicolon, New Line, Blank Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Nombre_Completo
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	FULLNBR
Field:	Z9NPRD
Datatype:	Number

Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	9702
Field:	Z9FPRD
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	"28/02/97"
Field:	Z4\$SMN
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	0
Field:	Z1PHRW
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4AN8
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""

Field:	Nombre_Completo_1
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	FULLNBR
Field:	Z4\$RFC
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$IMS
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$CBC
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstPDBA_P
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstGPA_P
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes

Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_P = 0; @Text(""); IstGPA_P); IstPDBA_D
Field:	
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstGPA_D
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_D = 0; @Text(""); IstGPA_D); IstPDBA_PV
Field:	
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstSaldoAnt_PV
Datatype:	Number

Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	IstGPA_PV + IstSaldoAct_PV; IstGPA_PV
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	Yes
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = 0; @Text(""); IstGPA_PV); IstSaldoAct_PV
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	Yes
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = 0; @Text(""); IstSaldoAct_PV); IstFOA
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	

Parenteses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstFOA = 0; @Text(""); IstFOA); totGPA_P
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parenteses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_P = ""; 0; @Sum(IstGPA_P)); totGPA_D
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parenteses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_D = ""; 0; @Sum(IstGPA_D)); totGPA_PV
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parenteses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	@If(IstGPA_PV = ""; 0; @Sum(IstGPA_PV));
Field:	NETO
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Sum(totGPA_P; totGPA_D; totGPA_PV);
Field:	Empleado_CN
Datatype:	Names
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Name([CN]; @UserName);
Field:	dtFirma
Datatype:	Time
Time Format:	02/12/97 07:52 PM
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(@IsDocBeingSaved & FIRMA = "Conforme" & Control != "Conforme"; @Do(@SetField("Control"; "Conforme"); @SetField("dtFirma"; @Now)); dtFirma);
Field:	FIRMA
Datatype:	Keywords
Input Multi-Value Separator(s):	Comma
Display Multi-Value Separator:	Comma
Help Description:	Firme si está conforme, de lo contrario solicite una aclaración. Una vez que se ha firmado no es posible cambiar este estatus
Field Type:	Computed when composed
Keyword User Interface:	Check Boxes
Allowable Keywords:	Conforme

Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""

Form:

Form Alias:
 Last Modification:
 Type:
 Window Title Formula:

Mensajes de Nómina

frmMsgNom
 26/11/97 01:31:22 PM
 Document
 SELECT @If(@IsNewDoc;
 "Nuevo Mensaje de
 Notificación de Nómina";
 "Mensaje de Notificación de
 Nómina");

Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:
 Field:

Yes
 Yes
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 [RH]
 [RH]
 MsgID

Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

Text
 [Not Assigned]
 Editable
 No
 Disabled
 No

Field:
 Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

Tema
 Text
 [Not Assigned]
 Editable
 No
 Disabled
 No

Field:
 Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

MSG
 Rich Text
 [Not Assigned]
 Editable
 No
 Disabled
 No

Form:

Form Alias:
 Last Modification:
 Type:
 Window Title Formula:

Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:

Field:

Datatype:
 Help Description:

Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

Field:

Datatype:
 Number Format:
 Percentage (value * 100)%:
 Parentheses on Negative Numbers:
 Punctuated at Thousands:
 Help Description:

Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:

Form:

Last Modification:
 Type:
 Include in Compose Menu:
 Include in Query by Form:

Registro de Usuario

frmCatUsuarios
 11/11/97 01:05:59 PM
 Document
 SELECT @If(@IsNewDoc;
 "Alta a un nuevo Registro";
 "Registro de " +
 @Text(catUserName));
 Yes
 Yes
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 No
 [None Assigned]
 All Users
 [RH]
 [Revisor]
 catUserName
 Names
 Teclee el identificador de
 usuario contenido en el ACL
 Editable
 No
 Disabled
 No
 catNoNomina
 Number
 General
 No
 No
 No
 No
 Teclee el número de Nómina
 que identifica al usuario
 Editable
 No
 Disabled
 No

Remitente_Destinatarario

02/12/97 05:13:41 PM
 Document
 No
 Yes

Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Field:	Remitente
Datatype:	Names
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Name([CN]; @UserName);
Field:	Fecha
Datatype:	Time
Time Format:	02/12/97 07:52 PM
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Created
Field:	SendTo
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon
Display Multi-Value Separator:	Semicolon
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""

Form:

Form Alias:	Respuesta a Aclaración
Last Modification:	Respuesta
Type:	02/12/97 07:48:01 PM
Window Title Formula:	Response to Response
	SELECT "Respuesta a
	Solicitud de Aclaración"
Include in Compose Menu:	No
Include in Query by Form:	Yes
Default Database Form:	No

Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	Yes
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Field:	\$Padre
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	\$Yo
Field:	resStatus
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	"Creada"
Field:	Revisor
Datatype:	Reader Names
Input Multi-Value Separator(s):	Comma, Semicolon
Display Multi-Value Separator:	Semicolon
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	Revisor
Field:	Empleado
Datatype:	Reader Names
Input Multi-Value Separator(s):	Comma
Display Multi-Value Separator:	Comma
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	Empleado
Field:	xZ9NPRD
Datatype:	Text

Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	xZ9NPRD
Field:	Tema
Datatype:	Text
Help Description:	Teclee en una línea un texto significativo
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Default Value Formula:	"Re: " + Tema;
Field:	Respuesta
Datatype:	Rich Text
Help Description:	Describe aquí el motivo de la aclaración
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	Origen
Datatype:	Rich Text
Help Description:	[Not Assigned]
Field Type:	Non-Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Form:

Last Modification:	sbfrmRespuesta
Type:	02/12/97 05:36:48 PM
Include in Compose Menu:	Document
Include in Query by Form:	No
Default Database Form:	Yes
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users

Form:

Form Alias:
 Last Modification:
 Type:
 Window Title Formula:

 Include in Compose Menu:
 Include in Query by Form:
 Default Database Form:
 Automatically Refresh Fields:
 Mail New Documents When Saving:
 Store Form In Documents:
 Inherit Default Field Values:
 Updates Become Responses:
 Retain Prior Versions As Responses:
 Activate Objects When Composing:
 Activate Objects When Editing:
 Activate Objects When Reading:
 Document Encryption Keys:
 Composed Documents May Be Read By:
 Form May Be Composed By:
 Field:
 Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:
 Field:
 Datatype:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:
 Field:
 Datatype:
 Input Multi-Value Separator(s):
 Display Multi-Value Separator:
 Help Description:
 Field Type:
 Sign When Mailing/In Section:
 Encryption:
 Update Requires Editor Access:
 Formula:
 Field:
 Datatype:
 Input Multi-Value Separator(s):

Solicitud de Aclaración

Aclaración
 02/12/97 07:48:30 PM
 Response to Document
 SELECT "Solicitud de
 Aclaración"
 No
 Yes
 No
 No
 No
 No
 No
 Yes
 No
 No
 No
 No
 [None Assigned]
 All Users
 All Users
 \$Padre
 Text
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 \$Yo
 Status
 Text
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 "Creada"
 Lector
 Reader Names
 Comma, Semicolon
 Semicolon
 [Not Assigned]
 Computed when composed
 No
 Disabled
 No
 ""
 Empleado
 Reader Names
 Comma

Display Multi-Value Separator:	Comma
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	Empleado
Field:	Remitente
Datatype:	Names
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Name([CN]; @UserName);
Field:	Fecha
Datatype:	Time
Time Format:	02/12/97 07:52 PM
Help Description:	[Not Assigned]
Field Type:	Computed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Created
Field:	SendTo
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon
Display Multi-Value Separator:	Semicolon
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	xZ9NPRD
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	Z9NPRD
Field:	Referencia
Datatype:	Rich Text
Help Description:	[Not Assigned]
Field Type:	Non-Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	\$Ref

Field:	Tema
Datatype:	Text
Help Description:	Teclee en una línea un texto significativo
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Field:	Comentario
Datatype:	Rich Text
Help Description:	Describe aquí el motivo de la aclaración
Field Type:	Editable
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Form:	Usuario
Last Modification:	17/10/97 05:41:12 PM
Type:	Document
Include in Compose Menu:	No
Include in Query by Form:	Yes
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	All Users
Field:	Usuario
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Name([CN]; @UserName);

Form:	(Perfil de Acceso)
Form Alias:	Perfil
Last Modification:	11/11/97 04:22:58 PM
Type:	Document
Window Title Formula:	SELECT "Variables de Configuración del Sistema"

Include in Compose Menu:	No
Include in Query by Form:	No
Default Database Form:	No
Automatically Refresh Fields:	No
Mail New Documents When Saving:	No
Store Form In Documents:	No
Inherit Default Field Values:	No
Updates Become Responses:	No
Retain Prior Versions As Responses:	No
Activate Objects When Composing:	No
Activate Objects When Editing:	No
Activate Objects When Reading:	No
Document Encryption Keys:	[None Assigned]
Composed Documents May Be Read By:	All Users
Form May Be Composed By:	[RH]
	[Revisor]
Field:	NotesUserName
Datatype:	Names
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	NotesUserName
Field:	AccessLevel
Datatype:	Keywords
Help Description:	Teclee el path donde se encuentra la información de nómina, v.gr. F:\InfoNomina\
	Editable
Field Type:	Radio Buttons
Keyword User Interface:	Manager
Allowable Keywords:	Designer
	Editor
	Author
	Reader
	Depositor
	No Access
	No
Sign When Mailing/In Section:	Disabled
Encryption:	No
Update Requires Editor Access:	"F:ODBC"
Default Value Formula:	@UpperCase(path);
Input Translation Formula:	Roles
Field:	Keywords
Datatype:	[Not Assigned]
Help Description:	Editable
Field Type:	Radio Buttons
Keyword User Interface:	[Empleado]
Allowable Keywords:	[RH]

Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Default Value Formula:

[Observador]
[Revisor]
No
Disabled
No
"[Empleado]"

Form:

Form Alias:
Last Modification:
Type:
Window Title Formula:

(Recibo)

Recibo
27/10/97 06:42:21 PM
Document
SELECT "Información de
Nómina de " +
@ProperCase(Z4\$NBR + " "
+ Z4\$APA + " " + Z4\$AMA);

Include in Compose Menu:
Include in Query by Form:
Default Database Form:
Automatically Refresh Fields:
Mail New Documents When Saving:
Store Form In Documents:
Inherit Default Field Values:
Updates Become Responses:
Retain Prior Versions As Responses:
Activate Objects When Composing:
Activate Objects When Editing:
Activate Objects When Reading:
Document Encryption Keys:
Composed Documents May Be Read By:
Form May Be Composed By:
Field:

No
Yes
No
No
No
No
No
No
No
No
No
No
No
No
No
No
[None Assigned]
All Users
All Users
Z9NPRD
Text

Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

[Not Assigned]
Computed when composed
No
Disabled
No
"9702"

Field:
Datatype:
Help Description:
Field Type:
Sign When Mailing/In Section:
Encryption:
Update Requires Editor Access:
Formula:

Z9FPRD
Text
[Not Assigned]
Computed when composed
No
Disabled
No
"28/02/97"

Field:
Datatype:
Number Format:
Percentage (value * 100)%:

Z4\$SMN
Number
Currency (2 Decimal Places)
No

Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	0
Field:	Z1PHRW
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4AN8
Datatype:	Number
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Nombre_Completo_1
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	FULLNBR
Field:	Z4\$RFC
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$IMS
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled

Update Requires Editor Access:	No
Formula:	""
Field:	Z4\$CBC
Datatype:	Text
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstPDBA_P
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstGPA_P
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_P = 0; @Text(""); IstGPA_P);
Field:	IstPDBA_D
Datatype:	Text
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	""
Field:	IstGPA_D
Datatype:	Number

Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_D = 0; @Text(""); IstGPA_D); IstPDBA_PV
Field:	Text
Datatype:	Comma, Semicolon, New Line
Input Multi-Value Separator(s):	New Line
Display Multi-Value Separator:	[Not Assigned]
Help Description:	Computed when composed
Field Type:	No
Sign When Mailing/In Section:	Disabled
Encryption:	No
Update Requires Editor Access:	No
Formula:	""
Field:	IstSaldoAnt_PV
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = 0; @Text(""); IstSaldoAnt); IstGPA_PV
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	Yes
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line

Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = 0; @Text(""); IstGPA_PV); IstSaldoAct_PV
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = 0; @Text(""); IstSaldoAct_); IstFOA
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Input Multi-Value Separator(s):	Comma, Semicolon, New Line
Display Multi-Value Separator:	New Line
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstFOA = 0; @Text(""); IstFOA); totGPA_P
Field:	Number
Datatype:	Currency (2 Decimal Places)
Number Format:	No
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No

Formula:	@If(IstGPA_P = ""; 0; @Sum(IstGPA_P));
Field:	totGPA_D
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_D = ""; 0; @Sum(IstGPA_D));
Field:	totGPA_PV
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@If(IstGPA_PV = ""; 0; @Sum(IstGPA_PV));
Field:	NETO
Datatype:	Number
Number Format:	Currency (2 Decimal Places)
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	Yes
Punctuated at Thousands:	Yes
Help Description:	[Not Assigned]
Field Type:	Computed when composed
Sign When Mailing/In Section:	No
Encryption:	Disabled
Update Requires Editor Access:	No
Formula:	@Sum(totGPA_P; totGPA_D; totGPA_PV);

View Information:

View Name:

Last Modification:
Default Database View:
No Response Hierarchy:
Categories Initially:
Unread Marks:

**Administración \ Catálogo
de Personal|vstRHCatPer**
12/11/97 02:17:18 PM
No
Disabled
Expanded
None

Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[Administración]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "frmCatUsuarios";
Column Number:	1
Column Title:	No. de Nómina
Column Formula:	@Text(catNoNomina);
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	Ascending
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Nombre
Column Formula:	@ProperCase(@Name([Abbreviate]; catUserName));
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
View Name:	Administración \
	Configuración vstAdmon
Last Modification:	12/11/97 02:17:34 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded

Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[Administración]
Time Relative Formulae:	No
Selection Formula:	SELECT @IsMember(Form; "frmConfiguración");
Column Number:	1
Column Title:	Fecha Modificación
Column Formula:	@Modified
Column Width:	13 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	Descending
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Fuente
Column Formula:	fuentes
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Ruta de acceso
Column Formula:	@If(fuente = "IBM AS/400"; biblioteca; path);
Column Width:	19 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No

Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No

View Name:

Last Modification:	Personal \ Aclaraciones 02/12/97 03:09:40 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal" & Control = "Aclaración" Form = "Aclaración";

Column Number:	1
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	Descending, Categorized
Justification:	Center
Totals:	None
Hide Detail Values:	No

Column Number:	2
Column Title:	No. de Nómina
Column Formula:	Z4AN8
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No

Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Column Width:	15 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	[Not Assigned]
Column Formula:	@Text(@Created) + " " + Tema;
Column Width:	1 Characters
List Separator:	
Hidden:	No
Responses Only:	Yes
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Aclaración
Column Formula:	""
Column Width:	14 Characters
List Separator:	
Hidden:	No
Responses Only:	No

Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	Tema
Column Formula:	""
Column Width:	15 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

View Name:	Personal \ Información de NóminalvstPERInfoHist
Last Modification:	01/12/97 05:36:26 PM
Default Database View:	Yes
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[Empleado]
Time Relative Formulae:	No
Selection Formula:	SELECT @IsMember(Form; "Principal");
Column Number:	1
Column Title:	Creación
Column Formula:	@Created
Column Width:	1 Characters
List Separator:	
Hidden:	Yes
Responses Only:	No
Icon:	No
Number Format:	General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	Descending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	11 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Firma
Column Formula:	@If(FIRMA = "Conforme"; 82; @If(Control = "Aclaración"; 122; 81));
Column Width:	3 Characters
List Separator:	
Hidden:	No

Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Firma
Column Formula:	dtFirma
Column Width:	11 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

	Personal	Mis
	Compromisos	
	Atrasados vstPERP	Pendientes
Last Modification:	17/11/97 04:01:00 PM	
Default Database View:	No	
No Response Hierarchy:	Disabled	
Categories Initially:	Expanded	
Unread Marks:	None	
Refresh Frequency:	Automatic	
Discard Index:	Never	
View May Be Used By:	[Empleado]	
Time Relative Formulae:	No	
Selection Formula:	SELECT (Form = "Principal" @AllDescendants) & (FIRMA != "Conforme");	
Column Number:	1	
Column Title:	[Not Assigned]	
Column Formula:	@Created	
Column Width:	1 Characters	
List Separator:		

Hidden:	Yes
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	Descending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Firma
Column Formula:	@If(Form = "Principal"; @If(FIRMA = "Conforme"; 82; @If(Control = "Aclaración"; 122; 81)); "");
Column Width:	3 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:38 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	4

Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	11 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	[Not Assigned]
Column Formula:	@Text(@Created) + " " + Tema;
Column Width:	1 Characters
List Separator:	
Hidden:	No
Responses Only:	Yes
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	Fecha de Aclaración
Column Formula:	""
Column Width:	13 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52 PM
Sorting:	No
Justification:	Center
Totals:	None

Hide Detail Values:	No
Column Number:	7
Column Title:	Tema
Column Formula:	""
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No

View Name: RH \ Compromisos
Atrasados|vstRHPendientes

Last Modification:	17/11/97 05:54:08 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[RH] [Observador]
Time Relative Formulae:	No
Selection Formula:	SELECT (Form = "Principal") & (FIRMA != "Conforme");
Column Number:	1
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Descending, Categorized
Justification:	Center
Totals:	None

Hide Detail Values:	No
Column Number:	2
Column Title:	Firma
Column Formula:	@If(Form = "Principal"; @If(FIRMA = "Conforme"; 82; @If(Control = "Aclaración"; 122; 81)); "");
Column Width:	3 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	No. Nómina
Column Formula:	Z4AN8
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No

Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No

View Name:

View Name:	RH \ Información x No. Nómina vstRHInfoxNNom
Last Modification:	02/12/97 04:53:39 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[RH] [Observador]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal";
Column Number:	1
Column Title:	No. Nómina / Nombre
Column Formula:	@Text(Z4AN8) + " " + @ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
	15 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Ascending, Categorized
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Firma
Column Formula:	@If(Control = "Aclaración"; 122; @If(FIRMA = "Conforme"; 82; 81));
	3 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

Last Modification:	RH \ Información x
Default Database View:	Periodo vstRHInfoxPeriodo
No Response Hierarchy:	02/12/97 04:53:24 PM
Categories Initially:	No
Unread Marks:	Disabled
	Expanded
	None

Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[RH] [Observador]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal";
Column Number:	1
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Ascending, Categorized
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	No. de Nómina
Column Formula:	Z4AN8
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Column Width:	20 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General

Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Firma
Column Formula:	@If(Control = "Aclaración"; 122; @If(FIRMA = "Conforme"; 82; 81)); 3 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
View Name:	RH \ Información x
Last Modification:	Persona vstRHInfoxPer 02/12/97 04:54:02 PM
Default Database View:	No
No Response Hierarchy:	Disabled

Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[RH]
	[Observador]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal";
Column Number:	1
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
	15 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Ascending, Categorized
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Firma
Column Formula:	@If(Control = "Aclaración";
	122; @If(FIRMA =
	"Conforme"; 82; 81));
	3 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	

Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	8 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

	RH \ Mensajes de
	Notificación vstMsgNom
Last Modification:	17/11/97 03:43:09 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[RH] [Observador]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "frmMsgNom";
Column Number:	1
Column Title:	ID de mensaje
Column Formula:	MsgID
Column Width:	8 Characters
List Separator:	
Hidden:	No

Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Ascending
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Asunto
Column Formula:	Tema
Column Width:	28 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No

View Name:

View Name:	RH \ Solicitudes de Aclaración vstRHSolxRevis ar
Last Modification:	02/12/97 01:06:59 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[Revisor]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal" & Control = "Aclaración" Form = "Aclaración";
Column Number:	1
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	
Hidden:	No

Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	Descending, Categorized
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	No. de Nómina
Column Formula:	Z4AN8
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Column Width:	15 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	[Not Assigned]
Column Formula:	@Text(@Created) + " " + Tema;

Column Width:	1 Characters
List Separator:	
Hidden:	No
Responses Only:	Yes
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Aclaración
Column Formula:	""
Column Width:	14 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	Tema
Column Formula:	""
Column Width:	15 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:39 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

	(Personal \ Mi último pago) vstPERInfoNom
Last Modification:	17/11/97 03:22:38 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	[Empleado]
Time Relative Formulae:	No
Selection Formula:	SELECT Form = "Principal";
Column Number:	1
Column Title:	Creación
Column Formula:	@Created
Column Width:	1 Characters
List Separator:	
Hidden:	Yes
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	Descending
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	10 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Fecha de Pago
Column Formula:	Z9FPRD
Column Width:	11 Characters

List Separator:	No
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52 PM
Sorting:	Ascending
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	Firma
Column Formula:	@If(FIRMA = "Conforme"; 82; @If(Control = "Aclaración"; 122; 81)); 3 Characters
Column Width:	
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	Yes
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Firma
Column Formula:	dtFirma
Column Width:	11 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

View Name:

	(Temporal) temporal
Last Modification:	02/12/97 07:50:32 PM
Default Database View:	No
No Response Hierarchy:	Disabled
Categories Initially:	Expanded
Unread Marks:	None
Refresh Frequency:	Automatic
Discard Index:	Never
View May Be Used By:	All Users
Time Relative Formulae:	No
Selection Formula:	SELECT @All;
Column Number:	1
Column Title:	Periodo
Column Formula:	Z9NPRD
Column Width:	5 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	Descending, Categorized
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	2
Column Title:	No. de Nómina
Column Formula:	Z4AN8
Column Width:	9 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	3
Column Title:	Nombre
Column Formula:	@ProperCase(Z4\$NBR + " " + Z4\$APA + " " + Z4\$AMA);
Column Width:	15 Characters

List Separator:	No
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	4
Column Title:	[Not Assigned]
Column Formula:	@Text(@Created) + " " + Tema;
Column Width:	1 Characters
List Separator:	
Hidden:	No
Responses Only:	Yes
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97
Sorting:	No
Justification:	Left
Totals:	None
Hide Detail Values:	No
Column Number:	5
Column Title:	Fecha de Aclaración
Column Formula:	""
Column Width:	14 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No
Column Number:	6
Column Title:	Tema

Column Formula:	""
Column Width:	15 Characters
List Separator:	
Hidden:	No
Responses Only:	No
Icon:	No
Number Format:	General
Percentage (value * 100)%:	No
Parentheses on Negative Numbers:	No
Punctuated at Thousands:	No
Time Format:	02/12/97 07:52:40 PM
Sorting:	No
Justification:	Center
Totals:	None
Hide Detail Values:	No

Agent Information:

Agent Name:	Crea en catálogo
Last Modification:	12/11/97 08:25:43 PM
Comment:	[Not Assigned]
Agent Name:	Despertador
Last Modification:	01/12/97 06:10:24 PM
Comment:	[Not Assigned]
Agent Name:	Enviar Notificaciones
Last Modification:	12/11/97 05:56:58 PM
Comment:	[Not Assigned]
Agent Name:	Enviar Recordatorio
Last Modification:	01/12/97 06:15:09 PM
Comment:	[Not Assigned]
Agent Name:	Importación Automática
Last Modification:	01/12/97 06:08:48 PM
Comment:	[Not Assigned]
Agent Name:	Importar Información
Last Modification:	02/12/97 12:08:37 PM
Comment:	[Not Assigned]
Agent Name:	Recordatorio Automático
Last Modification:	01/12/97 06:14:43 PM
Comment:	[Not Assigned]
Agent Name:	status importación
Last Modification:	13/11/97 10:48:48 AM
Comment:	[Not Assigned]

Stored Query Information:

[No Stored Queries Defined]

Script Library Information:

Script Library Information:

Anexo IV

Código en Lotus Script

Agente: Importar Información

Disparador: Manual

(Declarations)

```
Dim ilCont As Integer ' Contador empleado en iteraciones
Dim db As NotesDatabase ' Apuntador a la base de datos actual
Type DescNotes ' Se utiliza para almacenar los nombres de los campos en una
forma; relacionado con: ArregloCols()
    iTipoDato As Integer
    sNombreNotes As String
End Type

Type Ruta_de_Acceso ' Para cada archivo fuente existe una variable que almacenará la
ruta completa de acceso (incluyendo el nombre del archivo)
    InfoEmpleado As String
    Conceptos As String
    Detalle As String
    Descuentos As String
    FondoAhorro As String
    Depositos As String
    Control As String
End Type

Dim Ruta As Ruta_de_Acceso ' Esta variable almacenará la Ruta de Acceso a cada una de
las tablas o archivos para efectuar los Query's
Dim ArregloCols() As DescNotes ' Almacenará los nombres de los campos referentes a la
información del empleado para la asignación de los valores en un ciclo
Dim dITemp As Double, sITemp As String, vITemp As Variant
'Dim item As NotesItem
Dim igNPeriodo As Long ' Guarda el Número de Periodo de pago
Dim igFPago As String ' Guarda la fecha de pago

Type DBA
    IstPDBA As Variant
    IstGPA As Variant
End Type

Type typPVoluntarios
    IstPDBA As Variant
    IstSaldoAnt As Variant
    IstGPA As Variant
    IstSaldoAct As Variant
End Type
```

'Procedimiento principal

Sub Initialize

```
Dim session As New NotesSession
Dim acl As NotesACL
Dim entry As NotesACLEntry
Dim view As NotesView
Dim doc As NotesDocument

Set db = Session.CurrentDatabase
' Verifica si el usuario tiene el privilegio de ejecutar la importación de la nómina
Print "Verificando privilegios..."
Set acl = db.ACL
Set entry = acl.GetEntry( session.UserName )
If Not entry.IsRoleEnabled( "[RH]" ) Then
    Messagebox "No se cuenta con la autorización para ejecutar este tipo de acciones"
Else
    Print "Privilegios adecuados"
    respuesta%=Messagebox ("El proceso de Importación de la Nómina y generación de
avisos llevará unos minutos ¿desea continuar?", 4 + 32,"Aviso")
    If respuesta% = 6 Then
        Establecer_Conexion
    End If
End If
End Sub
```

' Se establece la conexión con el driver de ODBC

Sub Establecer_Conexion

```
Dim con As New ODBCConnection
Dim qryCtrl As New ODBCQuery
Dim resCtrl As New ODBCResultSet ' Almacena resultados del query al archivo de Control
Dim qry As New ODBCQuery
Dim res As New ODBCResultSet ' Almacena resultados del query al archivo de empleados
Dim ilStatus As Integer
Dim view As Notesview
Dim config As NotesDocument ' Apunta temporalmente al documento de configuración
Dim driver As String, user As String, pswd As String ' Almacenan el nombre del driver de
ODBC, el ID del usuario y el password, si es que existen
Dim path As String, ext As String ' Respectivamente almacenan la ruta (directorio o
biblioteca) de acceso a los archivos fuente y su extensión

' Revisión de la ubicación de la información fuente según el documento de configuración.
Print "Verificando ubicación de la información fuente..."
Set view = db.GetView( "vstAdmon" ) ' Acceso a la vista de Configuración
Set config=view.GetFirstDocument ' Acceso al documento de Configuración
If config.fuente(0)="IBM AS/400" Then ' Asignación del driver adecuado según el
documento de Configuración
    driver="B40"
    user="OCMTHJRM"
```

```

pswd="CORAZN"
path=config.biblioteca(0)
ext=""
Else
driver="Nómina"
user=""
pswd=""
path=config.path(0)
ext="." + config.ext(0)
End If

```

' Asignar la ruta de acceso para el Query; cada uno de los query's que se utilizarán en el orden en que aparecerán en el programa

```

Ruta.InfoEmpleado=path & config.arcInfoEmpleado(0) & ext ' Ruta de acceso para el
archivo de Información del Empleado
Ruta.Conceptos=path & "F069116" & ext ' F069116 es el archivo de
conceptos original de JDE
Ruta.Detalle=path & config.arcDetallePago(0) & ext ' Ruta de acceso para el
archivo que tiene los montos para cada concepto
Ruta.Descuentos=path & config.arcDescSaldo(0) & ext ' Ruta de acceso para el archivo
de Descuentos con Saldo
Ruta.FondoAhorro=path & config.arcFondoAhorro(0) & ext ' Ruta de acceso para el
archivo de Fondo de Ahorro
Ruta.Depositos=path & config.arcDepósitos(0) & ext ' Ruta de acceso para el
archivo de Depósitos
Ruta.Control=path & config.arcControl(0) & ext ' Ruta de acceso para el
archivo de Control

```

' Establecimiento de la conexión según la configuración encontrada.

Print "Estableciendo la conexión..."

If con.ConnectTo(driver, user, pswd) Then ' Conexión a la base de datos que contiene la información fuente

Print "Verificando archivo de control..."

Set qryCtrl.Connection = con

' Este query trae los parámetros de control como los son: Número de periodo, Fecha de Pago, Tipo de nómina y Status de Importación

qryCtrl.SQL = "SELECT * FROM " & Ruta.Control

Set resCtrl.Query = qryCtrl

If Not resCtrl.Execute() Then

Messagebox "Error: " & resCtrl.GetErrorMessage(DB_LASTERROR)

Else

Call resCtrl.NextRow

igNPeriodo=resCtrl.GetValue(1) ' Obtener el Número de Periodo de pago y asignarlo a una variable global para registro en los documentos

igFPago=resCtrl.GetValue(2) ' Obtener la fecha de pago y asignarla a una variable global para registro en los documentos

' Este query trae la información general de los empleados

Set qry.Connection = con

Print "Verificando archivo maestro de empleados..."

qry.SQL = "SELECT * FROM " & Ruta.InfoEmpleado

```

Set res.Query = qry
If Not res.Execute() Then
    MessageBox "Error: " & res.GetErrorMessage(DB_LASTERROR)
Else
    Call RegistroEmpleado(res,con)
End If
End If
ilStatus = con.Disconnect
' Actualizar la fecha y el periodo de la importación en el documento de configuración
Print "Actualizando documento de configuración..."
Set view = db.GetView( "vstAdmon" )
Set doc = view.GetFirstDocument
doc.dtlImportación=Date$          ' Registrar en el documento de configuración la fecha
de la última importación
doc.nPeriodo=igNPeriodo          ' Registrar en el documento de configuración el Número
del último Periodo importado
Call doc.Save (True, False)
Else
    MessageBox "No se pudo establecer la conexión", MB_ICONEXCLAMATION, "Error"
End If
End Sub

```

' Para cada empleado se genera un documento

Sub RegistroEmpleado(res As ODBCResultSet, con As ODBCConnection)

```

Dim tblDetalle As New ODBCResultSet
Dim qryDetalle As New ODBCQuery
Dim tblPVoluntarios As New ODBCResultSet
Dim qryPVoluntarios As New ODBCQuery
Dim tblDepositos As New ODBCResultSet
Dim qryDepositos As New ODBCQuery
Dim tblFAhorro As New ODBCResultSet
Dim qryFAhorro As New ODBCQuery
Dim Percepciones As DBA
Dim Deducciones As DBA
Dim PVoluntarios As typPVoluntarios
Dim IstFAhorro As Variant
Dim IstNames As Variant

Call CargaArreglo(ArregloCols())
Do
    Print "Efectuando conversión de la información fuente a formato Lotus Notes..."
    Call res.NextRow
    ' Este query trae los conceptos de percepciones y deducciones
    Set qryDetalle.Connection = con
    qryDetalle.SQL = "SELECT YCDISO, Z1GPA, Z1PHRW, YCEXA, YCPDBA FROM " &
Ruta.Detalle & ", " & Ruta.Conceptos & " WHERE Z1AN8="+ Str$(res.GetValue(3)) + " AND
Z1PDBA=YCPDBA"
    Set tblDetalle.Query = qryDetalle
    ' Este query trae la información de pagos voluntarios

```

```

Set qryPVoluntarios.Connection = con
qryPVoluntarios.SQL = "SELECT YCEXA, Z2BAD, Z2GPA, Z2AD, YCPDBA FROM " &
Ruta.Descuentos & ", " & Ruta.Conceptos & " WHERE Z2AN8="+ Str$(res.GetValue(3)) + "
AND Z2PDBA=YCPDBA"
Set tblPVoluntarios.Query = qryPVoluntarios
' Este query trae la información de Depósitos bancarios, los cuales serán anexados a las
deducciones
Set qryDepositos.Connection = con
qryDepositos.SQL = "SELECT * FROM " & Ruta.Depositos & " WHERE Z5AN8="+
Str$(res.GetValue(3))
Set tblDepositos.Query = qryDepositos

If Not tblDetalle.Execute() Then
    MessageBox "Error: " & tblDetalle.GetErrorMessage(DB_LASTERROR)
Else
    If Not tblPVoluntarios.Execute() Then
        MessageBox "Error: " & tblPVoluntarios.GetErrorMessage(DB_LASTERROR)
    Else
        If Not tblDepositos.Execute() Then
            MessageBox "Error: " & tblDepositos.GetErrorMessage(DB_LASTERROR)
        Else
            ' Esta rutina genera listas de conceptos y de importes para ser depositadas en los
campos correspondientes
            Call GeneraLista(tblDetalle, tblPVoluntarios, tblDepositos, Percepciones,
Deducciones, PVoluntarios)
            End If
        End If
    End If
End If

' Este query trae la información de Fondo de Ahorro
Set qryFAhorro.Connection = con
qryFAhorro.SQL = "SELECT Z3GPA, Z3RTAM FROM " & Ruta.FondoAhorro & "
WHERE (Z3PDBA=5654 OR Z3PDBA=5655) AND Z3AN8="+ Str$(res.GetValue(3))
Set tblFAhorro.Query = qryFAhorro
If Not tblFAhorro.Execute() Then
    MessageBox "Error: " & tblFAhorro.GetErrorMessage(DB_LASTERROR)
Else
    Redim IstFAhorro(1)
    IstFAhorro(0)=tblFAhorro.GetValue(1)
    Call tblFAhorro.NextRow
    IstFAhorro(1)=tblFAhorro.GetValue(1)
End If

' Busca si el documento que se va a generar ya existe, si sí, se reemplazan los valores,
de lo contrario se crea un documento nuevo
Set docExiste = LocalizaDoc(res, db, "Z4AN8","Z9NPRD")
If docExiste Is Nothing Then
    Print "Creando documento nuevo..."
    Set docExiste = New NotesDocument(db)
    docExiste.Form = "Principal"

```

```

' Datos Generales del empleado para documento nuevo
  For ilCont = 0 To Ubound(ArregloCols) ' ArregloCols contiene en un arreglo los
nombres de los campos de la forma
    Set item = docExiste.AppendItemValue(ArregloCols(ilCont).sNombreNotes,
res.GetValue(ilCont+1))
  Next
Else
  Print "Reemplazando documento existente..."
' Datos Generales del empleado para documento ya existente
  For ilCont = 0 To Ubound(ArregloCols)
    Set item = docExiste.ReplaceltemValue(ArregloCols(ilCont).sNombreNotes,
res.GetValue(ilCont+1))
  Next
End If
' Conceptos de Percepciones, Deducciones y Pagos Voluntarios con sus respectivos
importes:
docExiste.lstPDBA_P=Percepciones.lstPDBA
docExiste.lstGPA_P=Percepciones.lstGPA
docExiste.lstPDBA_D=Deducciones.lstPDBA
docExiste.lstGPA_D=Deducciones.lstGPA
docExiste.Z1PHRW=tblDetalle.GetValue(3)
docExiste.lstSaldoAnt_PV=PVoluntarios.lstSaldoAnt
docExiste.lstPDBA_PV = PVoluntarios.lstPDBA
docExiste.lstGPA_PV=PVoluntarios.lstGPA
docExiste.lstSaldoAct_PV=PVoluntarios.lstSaldoAct
docExiste.lstFOA=lstFAhorro
docExiste.Z9NPRD=igNPeriodo
docExiste.Z9FPRD=igFPago
docExiste.FIRMA=""
docExiste.Control="Pendiente"
docExiste.Z9NPRD=igNPeriodo
docExiste.Z9FPRD=igFPago
docExiste.Lectores=lstReaders(res, lstNames)
docExiste.Revisor=lstRevisores(res, lstNames)
docExiste.Empleado=Autor(res)

' Validar los campos Readers y Authors
Set Item = docExiste.GetFirstItem( "Lectores" )
Item.IsReaders=True
Item.IsNames=True
Set Item = docExiste.GetFirstItem( "Revisor" )
Item.IsReaders=True
Item.IsNames=True
Set Item = docExiste.GetFirstItem( "Empleado" )
Item.IsAuthors=True
Item.IsNames=True
Call docExiste.Save(True, False)
Loop Until res.IsEndofdata ' la tabla res contiene el ciclo mayor, las otras tablas se generan
por cada registro de ella
End Sub

```

Sub CargaArreglo(ArregloCols() As DescNotes)

Redim ArregloCols(10) As DescNotes

ArregloCols(0).sNombreNotes = "Z4HCMO"	' Nombre del Campo correspondiente a la Compañía
ArregloCols(1).sNombreNotes = "Z4HMCU"	' Nombre del Campo correspondiente al Centro de Costos
ArregloCols(2).sNombreNotes = "Z4AN8"	' Nombre del Campo correspondiente al Número de Nómina
ArregloCols(3).sNombreNotes = "Z4\$NBR"	' Nombre del Campo correspondiente al Nombre de pila
ArregloCols(4).sNombreNotes = "Z4\$APA"	' Nombre del Campo correspondiente al apellido paterno
ArregloCols(5).sNombreNotes = "Z4\$AMA"	' Nombre del Campo correspondiente al apellido materno
ArregloCols(6).sNombreNotes = "Z4\$SMN"	' Nombre del Campo correspondiente al sueldo mensual
ArregloCols(7).sNombreNotes = "Z4\$IMS"	' Nombre del Campo correspondiente al No. IMSS
ArregloCols(8).sNombreNotes = "Z4\$RFC"	' Nombre del Campo correspondiente al RFC
ArregloCols(9).sNombreNotes = "Z4\$FIG"	' Nombre del Campo correspondiente a la fecha de ingreso
ArregloCols(10).sNombreNotes = "Z4\$CBC"	' Nombre del Campo correspondiente a los #s de ctas bancarias.

End Sub

Function LocalizaDoc(res As ODBCResultset, db As NotesDatabase, Llave1 As String, Llave2 As String) As NotesDocument

```
Dim slSearch As String
Dim slTemp As String, slTemp2 As String
Dim collection As NotesDocumentCollection
Dim dateTime As New NotesDateTime("01/01/97")
slTemp=Cstr(res.GetValue(3))
slTemp2=Cstr(igNPeriodo)
slSearch = Llave1 + "=" & res.GetValue(3) & " & " & Llave2 & "=" & igNPeriodo
Set collection = db.Search(slSearch, dateTime, 0)
If Collection.Count >= 1 Then
    Set LocalizaDoc = Collection.GetFirstDocument
Else
    Set LocalizaDoc = Nothing
End If
End Function
```

**Sub Generalista(tblDetalle As ODBCResultSet, tblPVoluntarios As ODBCResultSet, tblDepositos As ODBCResultSet, _
Percepciones As DBA, Deducciones As DBA, PVoluntarios As typPVoluntarios)**

' Variables índices
Dim i,j,k As Integer

'Inicialización de los arreglos
Redim Percepciones.IstPDBA(0)
Redim Percepciones.IstGPA(0)
Redim Deducciones.IstPDBA(0)
Redim Deducciones.IstGPA(0)
Redim PVoluntarios.IstPDBA(0)
Redim PVoluntarios.IstSaldoAnt(0)
Redim PVoluntarios.IstGPA(0)
Redim PVoluntarios.IstSaldoAct(0)

i=-1
j=-1
k=-1

Do ' Recorrer la tabla de conceptos
Call tblDetalle.NextRow

' Creación de listas de conceptos e importes de percepciones y deducciones

If (tblDetalle.GetValue(1) = "P") Or (tblDetalle.GetValue(1) = "B") Then ' Identificar si el concepto actual es percepción o deducción

i=i+1

Redim Preserve Percepciones.IstPDBA(i)

Redim Preserve Percepciones.IstGPA(i)

Percepciones.IstPDBA(i)= tblDetalle.GetValue(4) ' Percepciones.IstPDBA es una lista de las percepciones

Percepciones.IstGPA(i)= tblDetalle.GetValue(2) ' Percepciones.IstPDBA es una lista de los montos de las percepciones

Else ' Es una deducción

j=j+1

Redim Preserve Deducciones.IstPDBA(j)

Redim Preserve Deducciones.IstGPA(j)

Deducciones.IstPDBA(j)= tblDetalle.GetValue(4) ' Deducciones.IstPDBA es una lista de las deducciones

Deducciones.IstGPA(j)= (-1)*Val(tblDetalle.GetValue(2)) ' Deducciones.IstPDBA es una lista de los montos de las deducciones

' Se multiplica por (-1) porque son deducciones y al final será sumadas con las percepciones en el front end

End If

Loop Until tblDetalle.IsEndofdata

' Creación de listas referentes a Pagos Voluntarios

Do ' Recorrer la tabla de Pagos Voluntarios

Call tblPVoluntarios.NextRow

k=k+1

```

Redim Preserve PVoluntarios.IstPDBA(k)
Redim Preserve PVoluntarios.IstSaldoAnt(k)
Redim Preserve PVoluntarios.IstGPA(k)
Redim Preserve PVoluntarios.IstSaldoAct(k)
PVoluntarios.IstPDBA(k)=tblPVoluntarios.GetValue(1) ' Acumula lista de conceptos
PVoluntarios.IstSaldoAnt(k)=Val(tblPVoluntarios.GetValue(4))+
Val(tblPVoluntarios.GetValue(3)) ' Acumula lista de saldos anteriores (por concepto)
PVoluntarios.IstGPA(k)=(-1)*Val(tblPVoluntarios.GetValue(3)) ' Acumula lista de importe o
aportación de este periodo
PVoluntarios.IstSaldoAct(k)=(-1)*Val(tblPVoluntarios.GetValue(4)) ' Acumula lista de saldo
actual por concepto
Loop Until tblPVoluntarios.IsEndofdata

```

```

' Los depósitos bancarios se agregan al final de la lista de deducciones

```

```

Do ' Recorrer la tabla de depósitos bancarios
Call tblDepositos.NextRow
j=j+1 ' Se incrementa j y no se inicializa debido a que se da continuidad a la lista
de deducciones
Redim Preserve Deducciones.IstPDBA(j) ' Se utilizan de nuevo las listas de deducciones, los
conceptos de depósitos bancarios se acumulan
Redim Preserve Deducciones.IstGPA(j) ' como saldo negativo al final de las deducciones
Select Case tblDepositos.GetValue(2) ' Según el tipo de cuenta es el mensaje que se
mostrará
Case "1" : Deducciones.IstPDBA(j)= "DEP. EN CTA. #" & tblDepositos.GetValue(3) ' Se
agrega el no. de cta. bancaria al concepto
Case "2" : Deducciones.IstPDBA(j)= "CHEQUE CTA. #" & tblDepositos.GetValue(3) ' Se
agrega el número de cuenta de cheques
Case "3" : Deducciones.IstPDBA(j)= "PAGO EN EFECTIVO"
Case Else : MessageBox "Valor no esperado", 48, "Error" ' Mostrar error y terminar ejecución
ante un valor no válido
End
End Select
Deducciones.IstGPA(j)=(-1)*Val(tblDepositos.GetValue(4)) ' Valor del depósito
Loop Until tblDepositos.IsEndofdata
End Sub

```

Function IstReaders(res As ODBCResultSet, IstNames As Variant) As Variant

```

Dim acl As NotesACL
Dim entry As NotesACLEntry
Dim rdCont As Integer
Dim view As Notesview
Redim IstNames(0)

rdCont=0
Set acl = db.ACL
Set entry = acl.GetFirstEntry
While Not ( entry Is Nothing )

```

```

    If ( entry.IsRoleEnabled( "[RH]" ) = True ) Then
        Redim Preserve IstNames(rdCont)
        IstNames(rdCont)=entry.Name
        rdCont=rdCont+1
    End If
    Set entry = acl.GetNextEntry( entry )
Wend

```

```

IstReaders=IstNames
End Function

```

' Obtiene una lista de nombres notes que tienen definido el rol de [Revisor] en el ACL

Function IstRevisores (res As ODBCResultSet, IstNames As Variant) As Variant

```

Dim acl As NotesACL
Dim entry As NotesACLEntry
Dim rdCont As Integer
Dim IstNames As Variant
Dim view As Notesview

Redim IstNames(0) ' Redimensionar el arreglo que contendrá los nombres
rdCont=0
' Recorrer el ACL
Set acl = db.ACL
Set entry = acl.GetFirstEntry
While Not ( entry Is Nothing )
    If ( entry.IsRoleEnabled( "[Revisor]" ) = True ) Then ' Si el nombre tiene el rol [Revisor]
entonces agregar a la lista
        Redim Preserve IstNames(rdCont)
        IstNames(rdCont)=entry.Name
        rdCont=rdCont+1
    End If
    Set entry = acl.GetNextEntry( entry )
Wend
IstRevisores=IstNames
End Function

```

'Obtiene el nombre notes del "dueño" del documento que se está generando, esto con la finalidad de depositarlo en un campo de tipo READERS

Function Autor (res As ODBCResultSet) As String

```

Dim view As Notesview
Dim formula As String

Set view = db.GetView( "vstRHCatPer" )
NoNomina$=Trim(Str$(res.GetValue(3)))
Set doc = view.GetDocumentByKey(NoNomina$)
Autor=doc.catUserName(0)
End Function

```

Agente: Enviar Recordatorio

Disparador: Manual

(Declarations)

Dim db As NotesDatabase

Sub Initialize

```
Dim session As New NotesSession
Dim view As Notesview
Dim doc, docMsg As NotesDocument
'Dim NoNomina As String
Dim slSubject, slMsg As String
```

```
Set db = Session.CurrentDatabase
' Obtener el IDENTIFICADOR del mensaje a enviar en el documento de configuración
Print "Verificando documento de configuración..."
```

```
Set view = db.GetView( "vstAdmon" )
```

```
Set doc = view.GetFirstDocument
```

```
MsgID=doc.msgRecordatorio(0)
```

```
' Obtener el documento que tiene el mensaje a enviar y almacenar temporalmente el Tema y el Mensaje
```

```
Print "Verificando mensaje a enviar..."
```

```
Set view = db.GetView( "vstMsgNom" )
```

```
Set docMsg = view.GetDocumentByKey(MsgID)
```

```
slSubject = docMsg.Tema
```

```
slMsg = docMsg.MSG
```

```
' Obtener la vista de documentos pendientes y el primer documento para enviar la referencia en el mail
```

```
Set view = db.GetView( "vstRHPendientes" )
```

```
Set doc = view.GetFirstDocument
```

```
While Not (doc Is Nothing) ' Para todos los documentos de la vista...
```

```
Print "Construyendo mensaje..."
```

```
Set docMsg = New NotesDocument(db) ' Se crea un nuevo documento para el envío de Memo
```

```
docMsg.SendTo=doc.Empleado ' doc.Empleado es el Nombre Notes del Empleado dueño del documento
```

```
docMsg.Subject=slSubject ' Se anexa el Tema
```

```
docMsg.Form="Memo" ' Se asigna forma
```

```
Set ritem = New NotesRichTextItem( docMsg, "Body" ) ' Se crea una referencia a un objeto para insertarse en el campo Body del Memo
```

```
ritem.AppendText(slMsg & " " & doc.Z9NPRD(0) & " ") ' Se anexa el mensaje y el número de periodo correspondiente
```

```
Call ritem.AppendDocLink( doc, "Documento Pendiente de firmar" ) ' Se anexa una referencia al documento
```

```
docMsg.Send(False) ' Se envía el Memo
```

```
Set doc = view.GetNextDocument(doc)
```

```
Wend
```

```
Print "Fin del proceso."
```

```
End Sub
```

Agente: Enviar Notificaciones

Disparador: Manual

(Declarations)

Dim db As NotesDatabase
Dim ritem As NotesRichTextItem

Sub Initialize

```
Dim view As Notesview
Dim doc As NotesDocument, docmail As NotesDocument
Dim acl As NotesACL
Dim entry As NotesACLEntry
Dim slSubject,slMsg As String
Dim slMsgID As String
Dim dlPeriodo As Double
Set db = Session.CurrentDatabase
Set acl = db.ACL
Set entry = acl.GetEntry( session.UserName )
If Not entry.IsRoleEnabled( "[RH]" ) Then
    Messagebox "No se cuenta con la autorización para ejecutar este tipo de acciones"
Else
' Obtener del documento de configuración el número de periodo de la última importación hecha
y el identificador del mensaje a enviar
    Set view = db.GetView( "vstAdmon" )
    Set doc = view.GetFirstDocument
    slMsgID=doc.msgNotificación(0)
    dlPeriodo=doc.nPeriodo(0)
' Obtener en base al identificador de mensaje el tema y el mensaje que se van a enviar
    Set view = db.GetView( "vstMsgNom" )
    Set doc = view.GetDocumentByKey(slMsgID)
    slSubject = doc.Tema
    slMsg = doc.MSG
' Para cada documento del periodo construir el mail que se va a enviar
    Set docmail = New NotesDocument(db)
    Set view = db.GetView( "vstRHInfoxPeriodo" )
    Set doc = view.GetFirstDocument
    While (Not doc Is Nothing)
        If doc.Z9NPRD(0) = dlPeriodo Then
            docmail.SendTo=doc.Empleado(0)
            docmail.Subject=slSubject
            docmail.Form="Memo"
            Set ritem = New NotesRichTextItem( docmail, "Body" )
            ritem.AppendText(slMsg & " #" & dlPeriodo & " ")
            Call ritem.AppendDocLink( doc, "Documento del periodo " & dlPeriodo )
            docmail.Send(False)
        End If
        Set doc = view.GetNextDocument(doc)
    Wend
End If
End Sub
```

Action: Enviar y Salir (Solicitud de Aclaración)

Sub Click(Source As Button)

Dim ws As New NotesUIWorkspace

Dim parent As NotesDocument, doc As NotesDocument, docCurrent As NotesDocument

Dim docUI As NotesUIDocument

Dim acl As NotesACL

Dim entry As NotesACLEntry

Dim rdCont As Integer

Dim lstSendTo As Variant

Dim flag As Variant

Dim Session As New NotesSession

Dim db As NotesDatabase

Redim lstSendTo(0)

rdCont=0

Set db = Session.CurrentDatabase

Set acl = db.ACL

Set entry = acl.GetFirstEntry

While Not (entry Is Nothing)

 If (entry.IsRoleEnabled("[Revisor]") = True) Then

 Redim Preserve lstSendTo(rdCont)

 lstSendTo(rdCont)=entry.Name

 rdCont=rdCont+1

 End If

 Set entry = acl.GetNextEntry(entry)

Wend

Set docUI = ws.CurrentDocument

Set docCurrent = docUI.Document

Set doc = New NotesDocument(db)

doc.SendTo=lstSendTo

doc.Subject="Aclaración: " & docCurrent.Tema(0)

doc.Form="Memo"

Set ritem = New NotesRichTextItem(doc, "Body")

ritem.AppendText("Una solicitud de aclaración fue enviada para el siguiente documento ")

Call ritem.AppendDocLink(docCurrent, "Aclaración")

doc.Send(False)

docCurrent.Status="Enviada"

doc.Lector=lstSendTo

Call docUI.Save

Call docUI.Close

Set parent = db.GetDocumentByUNID(docCurrent.ParentDocumentUNID)

parent.Control="Aclaración"

Call parent.Save (True, False)

End Sub

902873