

**UNIVERSIDAD DE GUAYAQUIL**

**Facultad de Ciencias Matemáticas y Físicas**

**Carrera de Ingeniería en Sistemas  
Computacionales**

**“Workflow para Petición de Bienes en una  
Facultad”**

**PROYECTO DE GRADO**

**Previo a la Obtención del Título de:**

**INGENIERO EN SISTEMAS COMPUTACIONALES**

**Autor(es):**

**ERIKA GAMBOA VICUÑA  
MARTHA MORENO CEDEÑO  
LÍA RUIZ LARREA**

**GUAYAQUIL - ECUADOR**

**Año: 2006**

## **AGRADECIMIENTO**

Nuestro más profundo agradecimiento en primer lugar es para el Todopoderoso que permitió llegáramos a esta etapa de nuestras vidas, dándonos sabiduría y fortaleza para alcanzar la meta trazada.

Queremos agradecer también a nuestros queridos padres, hermanos, demás familiares y amigos, que con su apoyo incondicional fueron un pilar fundamental para seguir adelante en nuestro camino.

También nuestro agradecimiento va para la Ing. Matilde Alvear, Directora de la Carrera, personal administrativo y todos los docentes que directa o indirectamente nos proporcionaron los conocimientos necesarios para convertirnos en profesionales altamente capacitados.

En general, nuestro agradecimiento es para todas aquellas personas que de una u otra forma contribuyeron para que pudiéramos llegar a la culminación de nuestro objetivo.

## **DEDICATORIA**

Este trabajo va dedicado en primer lugar a Jehová por haberme impulsado y brindado las fuerzas necesarias para alcanzar mis metas.

A mis padres Martha y Gabriel, por enseñarme que en la vida se debe luchar y tener tolerancia para llegar a los objetivos trazados.

A mi abuela y mis tíos, por guiarme y brindarme todo el apoyo para culminar otra etapa en mi vida.

A mi esposo Michael, por la ayuda constante y certera que me brindó en cada momento.

Finalmente a mi hija Arianita porque es ella la clave y la razón de éxito y superación.

**Erika C. Gamboa Vicuña**

## **DEDICATORIA**

Este proyecto va dedicado especialmente a Dios que me ha permitido llegar a esta etapa de mi vida, también a mis padres Mariana y Pedro, que han sido un pilar muy importante en mi vida, que con su apoyo y empuje hicieron que las cosas más difíciles de mi carrera no se vean de esa forma con sus sabios consejos y alientos pude seguir adelante.

A mis hermanos Carmen, Javier y en especial a Juan, gracias a la ayuda incondicional que me brindó desde el inicio de mi carrera, hizo posible que culminara esta etapa importante de mi vida estudiantil.

A mis amigos(as) ya que con ellos hemos luchado para seguir adelante y no rendirnos en este largo camino.

A mis familiares que directa o indirectamente me ayudaron en este proceso de mi vida.

**Martha Moreno Cedeño**

## **DEDICATORIA**

Dedico este trabajo en primer lugar a nuestro Padre Celestial que con su amor, me guió siempre y me dio la fortaleza necesaria para culminar mi objetivo.

De manera muy especial quiero dedicar este logro a mis padres, Ing. Victor Ruiz y Lic. Lilian Larrea, que con su apoyo incondicional, amor infinito y sabios consejos, supieron guiarme y darme fuerzas para seguir adelante; y a mi hermana Vicky, por ser también mi amiga y brindarme todo su apoyo.

Y para culminar, dedico este trabajo a mi amada y recordada abuelita, Emma Castello, que desde el cielo, me muestra siempre el camino correcto, y que con el amor y cuidados que me brindó durante toda su vida, supo hacer de mi una mujer de bien.

**Lía Ruiz Larrea**

# TRIBUNAL DE GRADUACIÓN

---

**PRESIDENTE DEL TRIBUNAL**

---

**PRIMER VOCAL**

---

**SEGUNDO VOCAL**

---

**SECRETARIO**

## DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a las suscritas, perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma”

(Reglamento de Graduación de la Carrera de Ingeniería en Sistemas computacionales, Art. 26)

---

Erika Gamboa Vicuña

---

Martha Moreno Cedeño

---

Lía Ruiz Larrea

# ÍNDICE GENERAL

<b>CAPITULO 1</b>	<b>1</b>
1 INTRODUCCIÓN	1
1.1 Temática	1
1.2 Solución	3
1.3 Definición del proyecto	4
1.4 Objetivos	5
1.4.1. Objetivos Generales	5
1.4.2. Objetivos Específicos	5
1.5. Alcance	6
1.6. Misión y Visión	7
1.6.1 Visión	7
1.6.2 Misión	7
1.7 Recursos	8
1.7.1 Hardware	8
1.7.2 Software	8
1.7.3 Humano	8
1.8 Metodología	9
1.8.1. Fase Análisis	9
1.8.2. Fase de Diseño	9

1.8.3. Fase de Desarrollo	10
1.8.4. Fase de Pruebas	10
<b>CAPITULO 2</b>	<b>11</b>
2 ANÁLISIS	11
2.1. Automatización De Flujos De Trabajo	11
2.1.1. ¿Qué es un Workflow?	11
2.1.2. Componentes de un Workflow	11
2.1.3. ¿Qué debe brindar un Workflow Workflow?	12
2.1.4. Beneficios de la automatización de flujos de trabajo	12
2.2 Plataforma Lotus Domino	13
2.2.1. ¿Qué es Lotus Domino?	13
2.2.2. ¿Qué es Notes/Domino?	14
2.2.3. Arquitectura Notes/Domino	14
2.2.4. Componentes Notes/Domino	15
2.2.5. Tipos de Servidores Domino	15
2.2.6. Tipos de Clientes	16
2.3. Almacenamiento de Datos	17
2.3.1. Modelo de contención de las Bases de Datos de Lotus Domino	17
2.3.2. Componentes de una Base de Datos Domino	18
2.3.3. Como almacena Lotus Domino la información	19
2.3.4. Características de las Notas	19

2.3.5. Cómo almacena los datos en una Nota	20
2.3.6. Tipos de Notas	20
2.3.7. Acceso a los datos de una nota	20
2.4 Seguridades	21
2.4.1 Seguridades del Servidor	21
2.4.2. Control de Acceso a Recursos	22
2.4.3. Seguridad de los Datos	24
2.5. Arquitectura de Mensajería Notes/Domino	25
2.5.1. Componentes del Ruteo de Correo	26
2.5.2. Topología de Ruteo	27
2.6 Análisis de los procesos involucrados	28
2.7. Funciones Del Workflow	32
2.7.1. Definiciones	32
2.8. Definiciones y Diagrama de Casos De Uso	34
2.8.1. Conceptos Básicos	34
2.8.2 Definiciones de los Casos de Usos	34
2.8.3 Diagrama de Caso de Uso	43
2.9 Modelo Conceptual	45
2.9.1 Conceptos Generales	45
2.10. Diagrama De Secuencia	46
2.11. Contratos	52

<b>CAPITULO 3</b>	64
3 DISEÑO Y DESARROLLO	64
3.1 Elementos de diseño	65
3.1.1 Tareas de las aplicaciones y los elementos de diseño correspondientes	66
3.1.2 Acceso a los elementos de diseño	66
3.1.3 Uso de formularios	67
3.1.4 Notas de bases de datos	68
3.1.5 Uso de las vistas	69
3.1.5.1 Características de una vista	70
3.1.5.2 Almacenamiento de las vistas	71
3.1.5.3 Creación de una vista	71
3.1.5.4 Facilidad de Indexamiento Notes	72
3.1.6 Uso de los recursos compartidos	73
3.1.6.1 Tipos de recursos compartidos	73
3.1.6.2 Ventajas de los recursos compartidos	74
3.1.7 Formularios relacionados	74
3.1.7.1 Tipos de formularios	75
3.1.8 Uso de acciones	77
3.1.8.1 Las acciones y los usuarios	78
3.1.9 Uso de los agentes	79
3.1.9.1 Acciones frente a agentes	80

3.2. Generalidades sobre Base de Datos	80
3.3. Diagrama de Clases	81
3.3.1 Conceptos básicos	81
3.3.2. Diagrama de clases	81
3.4 Desarrollo	83
3.4.1 Seguridades	83
3.4.2 Descripción General del Desarrollo	84
3.4.3 Códificación	86
<b>CAPITULO 4</b>	<b>87</b>
4 PRUEBAS	87
4.1 Generalidades	87
4.2 Pruebas realizadas	87
<b>CAPITULO 5</b>	<b>90</b>
5 RECOMENDACIONES Y CONCLUSIÓN	90
5.1 Recomendaciones	90
5.1.1 Hardware	90
5.1.2 Software	91
5.2 Conclusión	92

# ÍNDICE DE FIGURAS

Figura 2.1 Arquitectura Notes/Domino	14
Figura 2.2 Modelo de Contención de las bases de datos Domino	18
Figura 2.3 Componentes de una base de datos Domino	19
Figura 2.4 Niveles de Control de Acceso	23
Figura 2.5 Proceso de ruteo de correo	27
Figura 2.6 Gráfico de los Procesos Involucrados	31
Figura 2.7 Casos de Uso primera parte proceso	43
Figura 2.8 Casos de Uso segunda parte proceso	44
Figura 2.9 Representación del Concepto	45
Figura 2.10 Representación de Multiplicidad	46
Figura 2.11 Representación de Atributo	46
Figura 2.12 Diagramas de secuencia de Crear y Registrar Petición	47
Figura 2.13 Diagramas de secuencia de Aprobar Solicitud, Realizar Comparativo y Generar Solicitud de Compra	48
Figura 2.14 Diagramas de secuencia de Verificar Proveedores y Cotizaciones	49
Figura 2.15 Diagramas de secuencia de Registrar Despacho y Recibido	50
Figura 2.16 Diagramas de secuencia de Realizar Ingreso y Finalizar Trámite	51

Figura 3.1 Áreas del Domino Designer	64
Figura 3.2 Elementos de diseño de un formulario	68
Figura 3.3 Elementos de diseño de vista	70
Figura 3.4 Cómo crea Domino las vistas	72
Figura 3.5 Jerarquía de Respuesta	76
Figura 3.6 Diagrama de clases	82
Figura 3.7 Bases de datos	85

# ÍNDICE DE TABLAS

Tabla 1: Componentes Notes/Domino	15
Tabla 2: Tipos de Servidores Domino	16
Tabla 3: Tipos de clientes	17
Tabla 4: Control de Acceso a Recursos	22
Tabla 5: Niveles de Acceso	24
Tabla 6: Componentes de ruteo	26
Tabla 7: Funciones del Workflow	32
Tabla 8: Curso normal Crear Petición	35
Tabla 9: Curso normal Registrar Petición	36
Tabla 10: Curso normal Aprobar Solicitud	36
Tabla 11: Curso normal Realizar Comparativo y Generar Solicitud de Compra	37
Tabla 12: Curso normal Registrar Verificación de proveedores	38
Tabla 13: Curso normal Registrar Verificación cotizaciones	39
Tabla 14: Curso normal Registrar despacho de bienes	40
Tabla 15: Curso normal Registrar Recibido	41
Tabla 16: Curso normal Registrar Ingreso	42
Tabla 17: Curso normal Finalizar Trámite	42
Tabla 18: Elementos de Diseño	66

Tabla 19: Uso de formularios	67
Tabla 20: Tipos de notas	69
Tabla 21: Creación de una vista	72
Tabla 22: Asignación de roles y accesos	84

# CAPITULO 1

## 1. INTRODUCCIÓN

### 1.1. Temática

En la actualidad para realizar la petición de un bien que necesita una facultad se deben seguir una serie de pasos, tales como:

- Enviar la Solicitud de la petición al Decano.
- Una vez revisada la solicitud se la envía a Comisión Económica de la Facultad.
- La solicitud es enviada a Consejo Directivo de la Facultad.
- Una vez aprobada la solicitud, ésta es enviada a Administración Central para así realizar la verificación, si al revisar las cotizaciones adjuntas a la solicitud, se decide buscar una mejor propuesta para la compra, Administración Central se encargará de realizar dichas cotizaciones.

- Una vez revisada toda la documentación, se envía a División de Presupuesto para que se encargue de la verificación de los fondos propios.
- Revisados los fondos propios, la solicitud es enviada al Departamento Financiero para realizar la compra de los bienes necesarios.
- Al realizar la compra los bienes son entregados a Activos Fijos con la solicitud de la petición, se realiza la verificación de los bienes y se realiza el ingreso de los bienes al inventario.
- Se realiza la entrega de los bienes con la solicitud a Proveeduría, éste realiza la verificación de los bienes y comunica al administrador de la Facultad para que sean retirados.
- El administrador de la Facultad realiza la revisión pertinente de los bienes y los envía a Bodega.

En fin todo un largo y tedioso trabajo para el administrador de la Facultad hasta obtener una respuesta positiva de Administración Central, y así poder destinar el bien a quien corresponda para poder hacer uso del mismo, y esto sin contar con el tiempo que se pierde en el caso de que la solicitud sea negada o devuelta para realizar modificaciones específicas. Tomando en cuenta esta situación es necesario establecer un mecanismo mediante el cual todos los procesos antes mencionados no sean un

obstáculo al momento de realizar la petición de los bienes y así satisfacer las necesidades de la Facultad, para esto proponemos una solución de Automatización de dichos procesos

Esto permitirá tener un control de todos y cada uno de los bienes que son adquiridos para uso de la facultad.

## **1.2. Solución**

Después de haber observado y revisado la información obtenida por las personas involucradas en todas las actividades mencionadas en la temática, hemos definido que procesos automatizados serían la mejor solución para optimizar la realización de todos ellos.

Para llevar a cabo este proyecto se diseñará un workflow que incluirá cada uno de los pasos a realizar, y las personas responsables de cada actividad deberán ingresar a este workflow y registrar de alguna manera su aprobación para que se pueda avanzar a la siguiente actividad.

De esta manera se evitará la pérdida de tiempo y se optimizará el uso de los recursos utilizados para los mismos.

La herramienta escogida para llevar a cabo será Lotus Domino versión 6.5 por sus bondadosas opciones para el diseño y la facilidad que brinda para la automatización de procesos.

### **1.3. Definición del proyecto**

Este proyecto consiste en la realización de una herramienta que le permitirá a una facultad obtener bienes de una forma más rápida y tener un mayor control sobre éstos.

La forma como esto se logrará será a través de una automatización de los procesos que intervienen en la aprobación de las solicitudes, es decir el registro de la petición del bien por parte del solicitante hasta la última instancia que sería cuando el solicitante le da fin al trámite debido a que su solicitud fue atendida. En cada una de estas instancias se manejará correo electrónico por el cual se informará a las personas involucradas en estos procesos mediante un mensaje que indicará que deben ingresar al workflow para registrar su aprobación, de esta forma se agilizará el trámite que se le da a las mismas debido a que no será necesario que lleguen documentos impresos trasladados físicamente por una persona y que deban ser firmados, sino que automáticamente una vez aprobado un proceso, se podrá acceder al siguiente por la persona correspondiente al mismo y así hasta llegar a su última instancia.

Una vez realizada la compra, al administrador le deberá llegar un mail informando que puede proceder a retirar el bien solicitado. Luego el administrador registrará una firma de recibido en el workflow y esto

generará un mail que le llegará a la persona encargada de procesar el ingreso y su asignación.

El bodeguero de la facultad además utilizará el workflow para tener un control de los bienes existentes en la Facultad y a quienes han sido asignados, de tal forma que pueda manejar reportes detallados.

## **1.4. Objetivos**

### **1.4.1. Objetivos Generales**

- Proveer una herramienta útil a un administrador de una facultad para que tenga un control de los recursos que son asignados a la misma.
- Minimizar el tiempo que transcurre para la aprobación de una solicitud al momento de realizar la petición de un bien necesitado.
- Darle al administrador un mecanismo para que esté al tanto del trámite que se da a las solicitudes enviadas para que pueda conocer en qué momento se le da respuesta a éstas.

### **1.4.2. Objetivos Específicos**

- Planear, coordinar, controlar y evaluar las actividades relacionadas a la administración de los bienes que posee una facultad.

- Poseer un histórico de todos los bienes asignados a los distintos departamento de una facultad
- Permitir un fácil acceso al inventario de bienes de uso, su ubicación física, responsable, detalles de su adquisición.
- Informar a cada una de las personas que deban ingresar al workflow que tienen solicitudes por tramitar.

### **1.5. Alcance**

El Workflow permitirá realizar el registro, aprobación de solicitudes de bienes necesitados por una entidad. El egreso del bien se realizará para mantenimiento o para dar de baja a los mismos en caso de deterioro o traslado a otra facultad.

Permitirá el registro, verificación de fechas y disponibilidad de una forma inmediata de los bienes que se encuentran en bodega, así mismo la verificación de solicitudes aprobadas, negadas o devueltas de bienes requeridos y que deben ser comprados por medio de Administración Central, esto podrá ser manejado por el administrador mediante la intranet o Internet si la facultad posee una página Web. Las solicitudes serán enviadas a la entidad encargada de realizar las compras ya que el workflow no realiza este proceso. El workflow solo contemplará los procesos hasta el momento en que las peticiones salen de la facultad y se

envían a Administración Central, y los retoma en el momento en que el departamento de Activo Fijo envía un comunicado al administrador de la facultad informando que éste puede disponer de los bienes.

Será configurable para las distintas facultades en cuanto a los recursos o bienes que éstas utilizan, por ejemplo: computadores, accesorios médicos, muebles, etc.

## **1.6. Misión y Visión**

### **1.6.1. Misión**

Proveer una herramienta eficaz para la automatización de petición de bienes a una Facultad, permitiendo así tener una rápida respuesta y un mejor control sobre los mismos.

### **1.6.2. Visión**

Poder interactuar los diferentes workflows que utilicen Administración Central y las diferentes facultades para los procesos de peticiones de bienes, compras y adjudicación de los mismos, logrando así una total integración agilizando el trabajo de los diferentes departamentos involucrados en estos procesos.

## **1.7. Recursos**

### **1.7.1 Hardware**

1 PC con las siguientes características:

Procesador Pentium IV

Memoria 256 MB de RAM

Disco Duro de 40GB.

Dispositivos de Entrada/Salida: Monitor, Mouse, Teclado.

Equipos conectados a red LAN

### **1.7.2 Software**

Windows XP

Lotus Domino versión 6.5

Office 2003

### **1.7.3 Humano**

El equipo de desarrolladores que está formado por las integrantes de grupo, y tutor de la herramienta en la cual se va a desarrollar este proyecto, que nos ha dado el asesoramiento necesario para la realización del mismo.

## **1.8. Metodología**

Para nuestro proyecto hemos decidido utilizar la metodología UML en las etapas de Análisis y Diseño porque provee de métodos adecuados y eficaces para su ejecución, y en la etapa de Desarrollo se utilizará el modelo de Prototipos debido a que su uso provee una mayor interacción con los usuarios, obteniendo así una previa aceptación o rechazo de los mismos al trabajo realizado hasta el momento.

### **1.8.1. Fase Análisis**

En esta etapa de análisis realizaremos lo siguiente:

Plataforma Lotus Domino

Definición de Funciones del Workflow (Requerimiento)

Diagrama de Casos de Usos

Modelo Conceptual

Glosario

Diagrama de Secuencia

Contratos

### **1.8.2. Fase de Diseño**

El proceso del diseño se basa en lo siguiente:

Diagrama de Clases

### **1.8.3. Fase de Desarrollo**

Esta etapa consiste en la generación de código partiendo del diseño. La herramienta de desarrollo a utilizar será Lotus Domino versión 6.5 teniendo en cuenta las bondades que ésta presenta para el desarrollo del proyecto.

### **1.8.4. Fase de Pruebas**

Se realizarán las pruebas correspondientes conjuntamente con cada uno de los prototipos que serán presentados al usuario.

# CAPITULO 2

## 2. ANÁLISIS

### 2.1. Automatización De Flujos De Trabajo

#### 2.1.1. ¿Qué es un Workflow?

Es la integración de:

- La organización
- La información
- Los procesos

#### 2.1.2. Componentes de un Workflow

1. BIBLIOTECA DE PROCESOS LÓGICOS → Definición de los procesos, sus reglas de ruteo y secuencias.

2. DIRECTORIO DE LA ORGANIZACIÓN → Definición, agrupación y asignación de los participantes del workflow.
3. OBJETOS DE INFORMACIÓN → Composición gráfica de formularios y vías de acceso a la información a través de vistas.

### **2.1.3. ¿Qué debe brindar un Workflow Workflow?**

- Modelo electrónico de los procesos
- Eficiencia y estandarización de los procesos
- Monitoreo y control de las tareas (alarmas, leadtime)
- Reducir el manejo de papel y documentación
- Integración de la organización de los procesos (colaboración)
- Distribución de tareas automáticamente a los usuarios
- Simplificar las actividades manuales de los usuarios

### **2.1.4. Beneficios de la automatización de flujos de trabajo**

- Mayor eficiencia en las actividades diarias, reduciendo los ciclos de tiempos muertos de los empleados
- Mejora la productividad de cada participante dentro del flujo
- Mayor control de los procesos, identificación de los puntos de quiebre (balanceo de carga de trabajo)
- Ahorros de costos en gastos de papel y recursos de impresión, mediante la circulación de documentos electrónicos

- Colaboración más efectiva entre departamentos o compañías
- Proporciona la información correcta a la gente adecuada en el momento preciso (compartir conocimiento)
- Mejora los workflows de gestión de calidad

## **2.2 Plataforma Lotus Domino**

### **2.2.1. ¿Qué es Lotus Domino?**

Lotus Domino es un entorno de desarrollo rápido de aplicaciones que ofrece un gran beneficio sobre la inversión a sus usuarios.

Lotus Domino ofrece una gran variedad de servicios de colaboración y una infraestructura para las aplicaciones y el flujo de trabajo de procesos empresariales distribuidos. Lotus Domino e IBM WebSphere® incorporan puntos de integración en las áreas de servicio de directorio, seguridad, desarrollo de aplicaciones y Java™.

Entre las funciones que incluye Lotus Domino:

- Seguridad de Web integrada
- HTML nativo
- Eventos de JavaScripts™ integrados
- Integración con el workflow de archivos
- Servlets y agentes de Java
- Objetos distribuidos, como por ejemplo, CORBA/IIOP
- Integración de herramientas de diseño de otros fabricantes

- Integración con Microsoft y IIS
- Almacenamiento de objetos programables
- Replicación

Lotus Domino también se integra con orígenes de datos de servidor, como por ejemplo, IBM DB2® y ORACLE®, y con workflows de transacciones como IBM CICS®.

### 2.2.2. ¿Qué es Notes/Domino?

Los Servidores Domino trabajan con clientes Notes (y no Notes) para formar un ambiente integrado Cliente/Servidor. El ambiente Notes/Domino provee servicio que permiten a una organización ejecutar tareas de almacenamiento, comunicación e intercambio de información.

### 2.2.3. Arquitectura Notes/Domino

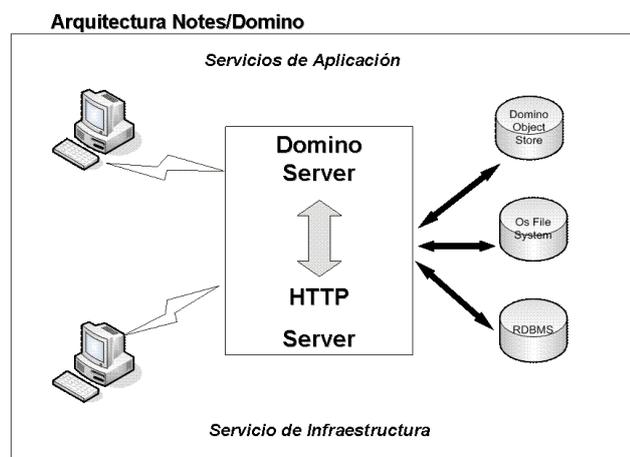


Figura 2.1

Fuente: Manual de Lotus Domino

### 2.2.4. Componentes Notes/Domino

Un ambiente Notes/Domino consiste de una combinación de los siguientes componentes cliente/servidor.

Tabla 1: Componentes Notes/Domino

Componente	Función
Servidor Domino	<ul style="list-style-type: none"> <li data-bbox="639 734 1377 869">➤ Un Servidor domino es un computador que corre el programa servidor Lotus Domino y almacena bases de datos Notes. Un servidor Domino corre servicios que manipulan datos Notes.</li> <li data-bbox="639 902 1377 1066">➤ Dependiendo de cual sea el requerimiento y quien sea el cliente el servidor puede extraer información de una variedad de fuentes, incluyendo objetos almacenados, el workflow de archivo, una base de datos relacional, o una combinación de los tres.</li> </ul>
Clientes Notes y Web	<ul style="list-style-type: none"> <li data-bbox="639 1075 1377 1171">➤ Los clientes Notes pueden acceder datos Domino en servidores y también localmente, suministrando accesos portátiles a los datos.</li> <li data-bbox="639 1205 1377 1267">➤ Clientes Web pueden acceder a datos Domino en el servidor para mostrarlos en un browser</li> </ul>

Fuente: Manual de Lotus Domino

### 2.2.5. Tipos de Servidores Domino

Cuando se instala un servidor Domino, se puede seleccionar entre tres tipos de servidores, los cuales se describe a continuación:

Tabla 2: Tipos de Servidores Domino

Tipo de Servidor	Función
Domino Utility Server	Provee estándares de servicios de aplicaciones Domino y bases de datos Domino personalizadas para clientes Notes y Web, además del soporte para respaldo.  <b>Nota:</b> No incluye servicios de Mensajería
Domino Messaging Server	Provee servicios de mensajería  <b>Nota:</b> No incluye servicios de aplicaciones
Domino Enterprise Server	Incluye las funcionalidades de los dos anteriores, incluyendo el soporte para el respaldo.

Fuente: Manual de Lotus Domino

### 2.2.6. Tipos de Clientes

Los usuarios que tienen archivos de correo en un Servidor Domino pueden usar el Cliente Notes o el Cliente de correos de internet, para acceder a su correo.

- **Cliente Notes** → Usan protocolos Notes para enviar e ingresar al correo en un servidor domino; un cliente notes puede además actuar como un Cliente de correos de internet
- **Cliente de correos de internet** → Ingresa a los archivos de correo a través del domino POP3, IMAP, o Servidores HTTP. Clientes POP3 e IMAP envían correos usando SMTP.

La siguiente tabla muestra los tipos de clientes disponibles con servidores domino:

Tabla 3: Tipos de clientes

Cliente	Funciones
<b>Cientes Notes:</b> <ul style="list-style-type: none"> <li>➤ Notes</li> <li>➤ IBM® Lotus® Domino Designer®</li> <li>➤ Administrador Lotus Domino</li> </ul>	<p>Permite a los usuarios acceder a base de datos Notes en un servidor domino, correo, navegador web.</p> <p>Soporta la creación, y desarrollo de bases de datos o aplicaciones domino (este cliente no es usado para acceder al correo).</p> <p>Permite a los usuarios con privilegios de administrador ejecutar tareas administrativas en un ambiente domino (este cliente no es usado para acceder al correo)</p>
<b>Cientes de Internet:</b> <ul style="list-style-type: none"> <li>➤ IBM® Lotus® iNotes™ Web Access</li> <li>➤ Web</li> <li>➤ POP3</li> <li>➤ IMAP</li> </ul>	<p>Provee acceso dentro o fuera de línea a la mensajería del corazón de Domino, colaboración, y funciones de administración de información personal a través del navegador Web.</p> <p>Soporta acceso de correo y calendario para navegadores Web.</p> <p>Permite el acceso a correo de un servidor POP3 flexible</p> <p>Soporta el acceso de correo a un servidor IMAP habilitado.</p>

Fuente: Manual de Lotus Domino

## 2.3. Almacenamiento de Datos

### 2.3.1. Modelo de contención de las Bases de Datos de Lotus Domino

Las bases de datos Domino son archivos que se sitúan en el workflow de archivos. Lotus Domino utiliza la ruta de acceso y el nombre de archivo para identificar la base de datos.

Este tipo de relación se denomina modelo de contención. El workflow de archivos contiene base de datos, las bases de datos contienen notas y las notas contienen información.

### Modelo de Contención de las bases de datos Domino

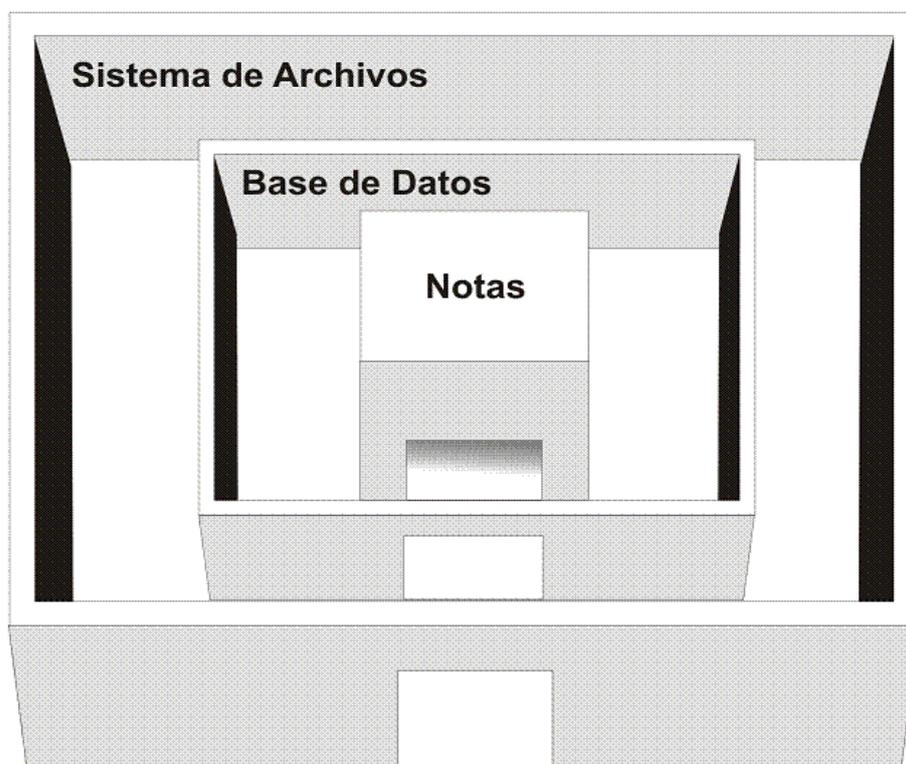


Figura 2.2

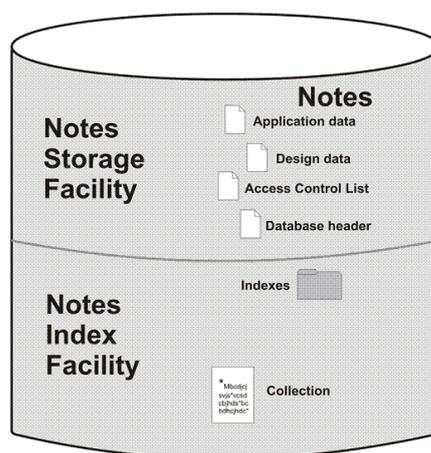
Fuente: Manual de Lotus Domino

### 2.3.2. Componentes de una Base de Datos Domino

Contiene dos áreas de funcionalidad:

- **Notes Storage Facility (NSF)** mantiene los datos y los elementos de diseño de la aplicación.

- **Notes Index Facility (NIF)** crea y mantiene listas para localizar los datos en la aplicación.



Componentes de una base de datos Domino

Figura 2.3

Fuente: Manual de Lotus Domino

### 2.3.3. Como almacena Lotus Domino la información

La nota es la estructura de datos utilizada para almacenar toda la información en una base de datos de Notes o de Domino.

### 2.3.4. Características de las Notas

Todas las notas de una base de datos son esencialmente iguales, son semejantes entre sí y no mantienen ni relaciones jerárquicas ni ningún otro tipo de relación. Las notas únicamente difieren en el tipo de valores de datos que contienen y en el uso que Lotus Domino hace de ellas.

### **2.3.5. Cómo almacena los datos en una Nota**

Una nota almacena los datos en forma de **elementos**. Cada elemento se identifica mediante un nombre y contiene una lista de valores. Los valores de cada elemento deben ser del mismo tipo de datos. Los elementos también almacenan información sobre los datos. Estos **metadatos** están a disposición de la aplicación y de los servicios de Domino. Los metadatos son datos sobre datos.

### **2.3.6. Tipos de Notas**

Una base de datos Domino contiene notas de encabezado, de lista de control de acceso (ACL) y de diseño. También incluye notas de datos desde el momento en que los usuarios empiezan a crearlas. Los usuarios y desarrolladores crean ciertos tipos de notas mientras que Lotus Domino crea otras para controlar y realizar seguimiento de la información en la base de datos.

### **2.3.7. Acceso a los datos de una nota**

Los datos se almacenan en un tipo específico de nota, denominado **nota de datos o documentos**.

#### **Otras formas de acceder a los datos**

Domino proporciona otras formas de acceder a los datos entre los que se incluyen:

- Los campos de un formulario
- Las columnas de una vista
- Mediante programación, a través de los siguientes elementos:
  - Lenguajes de Fórmula
  - LotusScript®
  - JavaScript
  - Java
  - XML

## **2.4 Seguridades**

### **2.4.1 Seguridades del Servidor**

Los controles de seguridad determinan el acceso a los servidores y recursos en el ambiente Domino.

El uso de controles permite el acceso a usuarios y servidores autorizados y bloquea el acceso para usuarios y servidores no identificados o específicos.

Existen dos niveles de seguridad:

- La autenticación. Establece el acceso entre dos entidades mediante una certificación.
- Control de acceso. Determina que información está habilitada para una entidad, la misma que puede ser un cliente o un servidor.

### 2.4.2. Control de Acceso a Recursos

La plataforma Domino provee un selectivo control del acceso a los recursos del servidor, pueden ser localizados en diferentes niveles: el servidor, base de datos (incluyendo la información de los campos de un formulario), agentes, applets y paginas Web.

El uso del control depende del nivel de seguridad requerido por las aplicaciones y el acceso de los usuarios.

Tabla 4: Control de Acceso a Recursos

Acceso a	Es controlado por
Servidores, incluyendo clientes Notes y clientes Web, y otros servidores Domino.	El seteo y restricciones del servidor
Carpetas de archivos Domino	Controles de acceso y restricciones
Ejecución de Applets en Java	Las restricciones del servidor
Ejecución de Agentes Domino	Las restricciones del servidor
Páginas Web	Controles del servidor Web
Base de Datos <ul style="list-style-type: none"> <li>➤ Formularios y vistas</li> <li>➤ Documentos</li> <li>➤ Campos</li> </ul>	<ul style="list-style-type: none"> <li>➤ Lista de control de Acceso (LCA)</li> <li>➤ Grupos</li> <li>➤ Roles</li> <li>➤ Encriptación para control de campos.</li> </ul>

Fuente: Manual de Lotus Domino

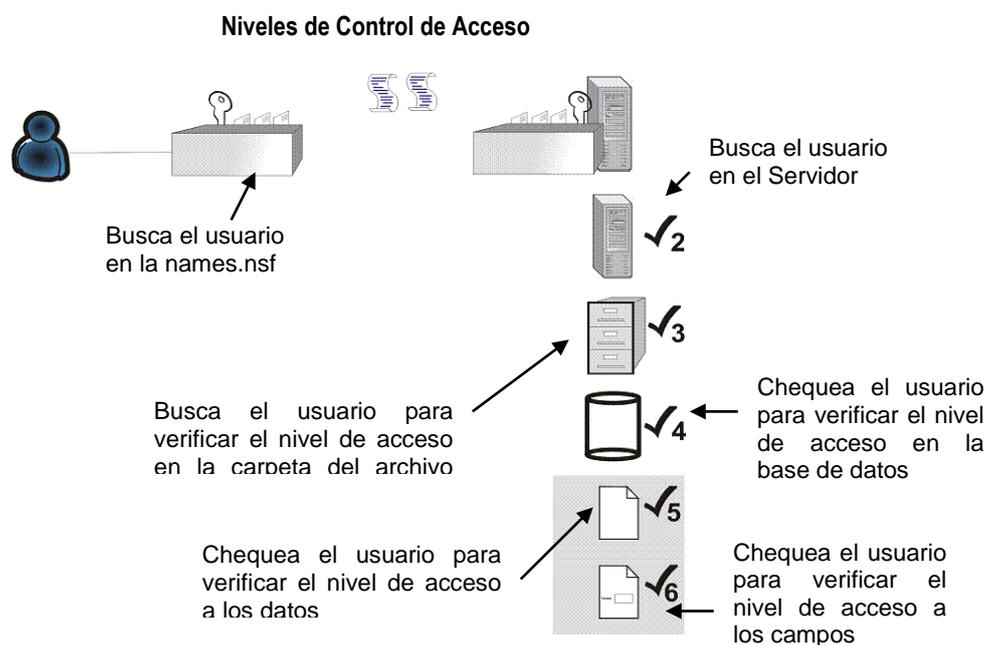


Figura 2.4

Fuente: Manual de Lotus Domino

La figura anterior describe gráficamente los estados del control de acceso para componentes Domino:

1. Búsqueda en la names.nsf
2. Búsqueda del nombre de la persona, debe estar chequeado en la parte de seguridad del documento del servidor
3. Nivel de archivo permitido, acceso a carpetas o archivos
4. Define acceso a BD
5. Verificación de acceso a datos
6. Encriptación, verificación de poder desenscriptar

### 2.4.3. Seguridad de los Datos

**Lista de Control de Acceso (LCA).**- Determina el tipo de acceso que se da a la base de datos.

Tabla 5: Niveles de Acceso

<b>Nivel</b>	<b>Acceso del Usuario</b>	<b>Acceso del Servidor</b>
<b>Sin acceso</b>	No tiene acceso a la base de datos	No tiene acceso a la base de datos (excepto, opcionalmente para una clase especial de documentos llamados documentos públicos).
<b>Depositador</b>	Puede crear documentos en la base de datos, pero no leerlos, editarlos o eliminarlos incluyendo los que él mismo crea.	No permite replicación. Nota: Este nivel no es normalmente a servidores.
<b>Lector</b>	Puede leer documentos, pero no crear, editar o eliminarlos.	Puede replicar solo para recibir (no envía documentos.)
<b>Autor</b>	Puede crear y leer documentos, y puede editar documentos creados por el si se define los campos usados como tipo authors.	Puede replicar documentos nuevos, pero no modificarlos. Nota: Este nivel no es normalmente a servidores.
<b>Editor</b>	Pueden crear, leer y editar todos los documentos.	Puede replicar todos los documentos nuevos y los cambiados.
<b>Diseñador</b>	Puede modificar el diseño de la base de datos pero no modificar la LCA o borrar la base de datos.	Puede replicar todos los documentos nuevos y los cambiados, y replicar elementos de diseño.
<b>Administrador</b>	Pueden ejecutar todas las operaciones de la base de datos, incluyendo cambios en la LCA y eliminar la base de datos	Puede replicar tanto los cambios en la LCA como cambios de diseño y documentos.

Fuente: Manual de Lotus Domino

## 2.5. Arquitectura de Mensajería Notes/Domino

Tiene como objetivo:

- Identificar los componentes para el ruteo de mails
- Describir como trabaja el ruteo de mails
- Diseñar una topología de ruteo de mails

El ruteo de mail ocurre automáticamente entre servidores de igual DNN (Domino Name Network).

Para habilitar la comunicación entre servidores de diferentes DNN se debe de configurar el documento de conexión, incluyendo la información específica de conexión tal como definiciones de servidor, entrega de requerimientos calendarizados y tamaño de cola de mensajes.

El ruteo se lo puede realizar a través de las siguientes opciones de protocolo existentes:

- **NRPC → Llamada a Procedimientos Remotos Notes**, pueden ser configurados para el ruteo de correo en un dominio Domino y entre dominios Domino. Trabaja a través del Puerto 1352
- **SMTP → Protocolo Simple de Transferencia de Mensajes**, es un protocolo de ruteo estándar de Internet el cual es nativo en Domino, soporta solamente el protocolo TCP/IP. Trabaja a través del puerto 25.

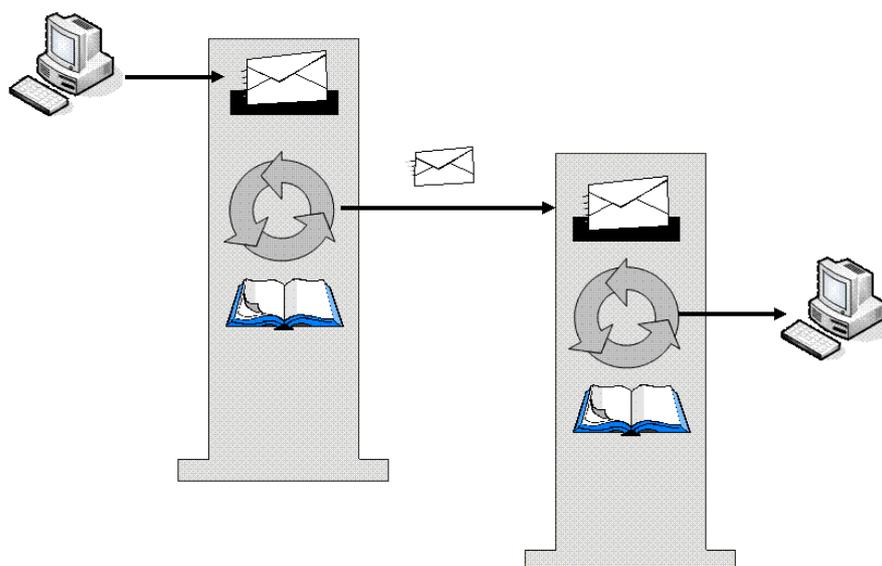
### 2.5.1. Componentes del Ruteo de Correo

El ruteo de mail es una de las características claves para algunas implementaciones domino. Los archivos y tareas de correo Domino trabajan juntos para proveer un ambiente de mensajería consistente y confiable.

Tabla 6: Componentes de ruteo

Componente	Definición
<b>Archivos de Correo</b>	Base de datos domino en la cual el usuario crea, envía, recupera y almacena mensajes de correo
<b>Servidor de Correo</b>	Es el servidor donde los archivos de correo residen y están especificados en el documento de personas en el directorio domino
<b>Verificador de correo (Mailer)</b>	Reside en la estación de trabajo y ejecuta las siguientes tareas: <ul style="list-style-type: none"> <li>➤ Verifica la existencia del nombre en el listado del Directorio Domino.</li> <li>➤ Convierte el mensaje a una extensión de correo de Internet multipropósito (MIME) si es necesario.</li> <li>➤ Deposita el mensaje en la caja de correo del servidor de correo del que envía.</li> </ul>
<b>Directorio Domino</b>	Es la base de datos que almacena la información acerca de el servidor de correo que envía (posiblemente los destinatarios), workflow de archivos de correo, nombre de archivos de correo, dirección de correo, y conexiones con otros servidores para la transferencia y entrega.
<b>Caja de correo</b>	Una base de datos especial que reside en cada servidor usada para la entrega de correos. El correo es temporalmente almacenado en la caja de correo antes que el router envíe o transfiera el correo.
<b>Router</b>	Tarea básica del servidor que libera y transfiere correos. Chequea el Directorio Domino para las conexiones con otros servidores y deposita correos en los archivos de correo de los usuarios y en las cajas de correo de otros servidores.

Fuente: Manual de Lotus Domino



### Proceso de Ruteo de Correo

Figura 2.5

Fuente: Manual de Lotus Domino

#### 2.5.2. Topología de Ruteo

La topología permite establecer que servidores están conectados y como ellos comunican la información específica.

Una topología define como los servidores de correo son configurados dentro de una organización. Los tipos de topología varían dependiendo el tamaño y tipo de organización:

- **Organizaciones Pequeñas** (4 o menos servidores) → Usa la topología Punto a Punto (Peer to Peer) para el ruteo de correo.
- **Organizaciones Medianas** (4 a 6 servidores) → Usa la topología Punto a Punto (Peer to Peer) combinada con la topología Hub-and-Spoke, para el ruteo de correo.

- **Organizaciones Grandes** (6 o más servidores)→ Usa la topología Hub-and-Spoke, para el ruteo de correo.

## 2.6 Análisis de los procesos involucrados

El usuario **Secretaria** crea una solicitud de petición. Una vez creada dicha solicitud, el Workflow debe notificar vía electrónica (mail) sobre la nueva solicitud y dirigirla para que la apruebe el **Director**.

El **Director** puede adoptar dos decisiones: **Aprobar** la solicitud o **Rechazarla**, si el **Director** la rechaza, la solicitud pasa a Estado rechazado y es almacenada para mantener un listado histórico. En caso de aprobarla se debe notificar vía electrónica sobre la solicitud y dirigirla al **Decano** para que la revise y dé inicio al trámite.

El **Decano** revisa la solicitud, pone a conocimiento de **Comisión Económica** (Decano, Directores, Administrador, Custodio de la Cuenta), y del **H. Consejo Directivo** (Decano, Subdecano, Directores, Administrador, Representante de los Trabajadores, Delegado Estudiantil, Vocales) para que indiquen si hay o no disponibilidad presupuestaria para la adquisición.

Si no hay disponibilidad, el **Decano** notifica al **Director** a través de un mail que su solicitud ha sido rechazada porque el departamento no cuenta con el presupuesto, y se almacena para mantener un listado histórico.

Si es aprobada el **Decano** registra la aprobación de la solicitud y notifica al **Administrador** vía mail, que existe una nueva solicitud de petición.

El **Administrador** (o su secretaria **Secre Admin**) revisa la solicitud de petición, elabora el **Cuadro Comparativo de Cotizaciones**, y adjunta las cotizaciones que ya previamente ha recibido de diferentes proveedores y escaneado. Una vez realizado el cuadro, genera una **Solicitud de Compra** con la autorización del Decano que estará vinculada a la **Solicitud de Petición**. Luego, el **Administrador** envía la documentación a través del workflow quien notifica a **Vicerrectorado Administrativo** que existe una documentación por verificar.

**Vicerrectorado Administrativo** registra el estado de verificación en la solicitud y dirige la documentación al **Jefe de Adquisiciones**, el workflow notifica que existe la documentación por verificar, o rechaza la solicitud y se notifica al Decano.

El **Jefe de Adquisiciones** revisa toda la documentación (cotizaciones y cuadro comparativo), entonces da su aprobación y se notifica al rol correspondiente para que continúe el trámite (Workflow Compras).

Una vez registrados los resultados del Workflow de Compras, el workflow notifica al **Jefe de Activo Fijo** sobre los bienes que han sido adquiridos.

El **Jefe de Activo Fijo** ingresa al workflow para registrar el recibido, codifica e ingresa la información sobre los bienes, luego notifica al **Administrador** de la facultad correspondiente a través del workflow acerca del despacho de los bienes adquiridos.

El **Administrador** recibe la notificación y retira los bienes verificados junto con el personal delegado por el **Departamento de Activo Fijo**. Una vez retirados los bienes, el Administrador ingresa al workflow para registrar el recibido de dichos bienes, y el workflow notifica al **Bodeguero** acerca del ingreso de los bienes a la facultad.

El **Bodeguero** realiza el ingreso de los bienes al workflow. Una vez ingresados los bienes el workflow notifica al **Director** solicitante acerca de los bienes ya registrados.

El **Director** retira los bienes solicitados, e ingresa al workflow para registrar el recibido de los bienes, automáticamente el workflow notifica al **Administrador** sobre el cambio de estado (fin del trámite) de la solicitud y el recibido de los bienes por parte del **Director**.

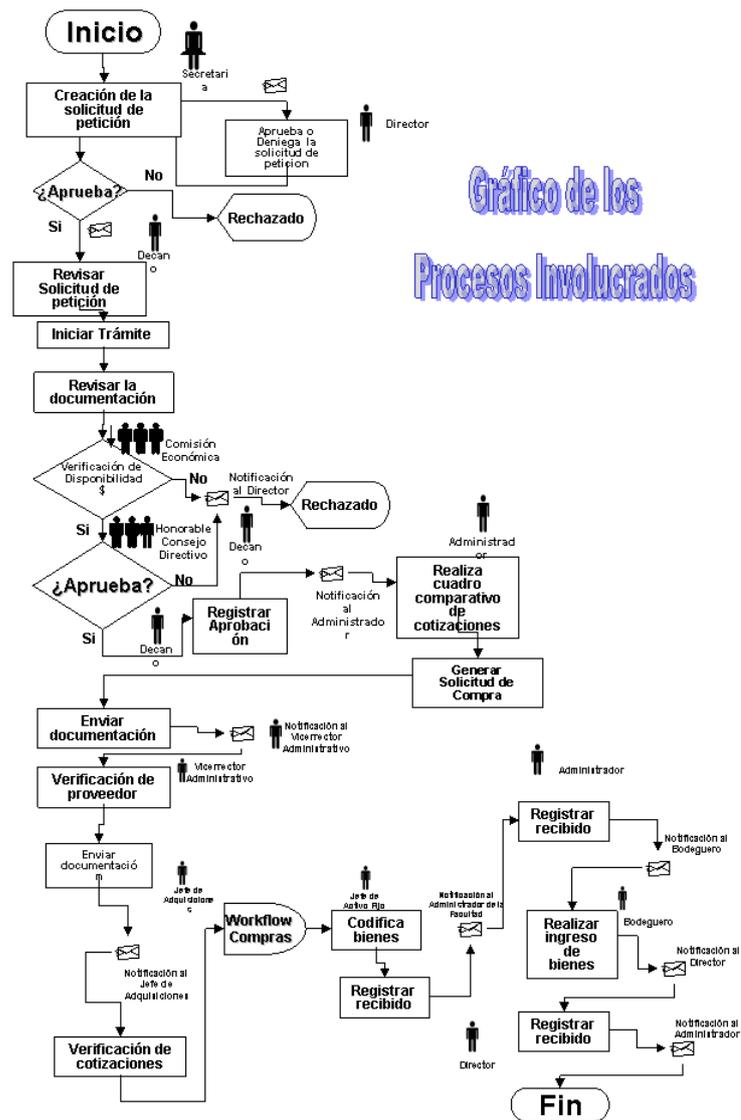


Figura 2.6

Fuente: Grupo de Proyecto

## 2.7. Funciones Del Workflow

### 2.7.1. Definiciones

**Funciones del Workflow.** Son lo que éste habrá de hacer, hay que identificarlas y listarlas en grupos cohesivos y lógicos.

**Categorías de las funciones.** Se clasifican para establecer una prioridad entre ellas e identificar las que de lo contrario pasarían inadvertidas. Las categorías son:

- **Evidentes:** Deben realizarse, y el usuario debería saber que se ha realizado.
- **Ocultas:** Debe realizarse, aunque no es visible para el usuario.
- **Superfluas:** Opcionales.

**Atributos.** Son características o dimensiones; no son funciones. Por ejemplo: Tiempo de respuesta, tolerancia a fallas, etc.

**Detalles de atributos:** Valores discretos, confusos o simbólicos.

**Restricciones de frontera:** Condiciones obligatorias de frontera, generalmente en un rango numérico de los valores de un atributo

Tabla 7: Funciones del Workflow

Ref.	Función	Categoría	Atributo	Detalles y Restricciones	Categoría
R1.1	Realizar nueva solicitud de petición de bienes.	Evidente	Facilidad de uso	(Det) El usuario solo deberá llenar un formulario con los datos de los bienes solicitados y grabarlo.	Obligatorio
			Metáfora de interfaz	(Det) Pantalla amigable (Det)Colorido	Obligatorio Opcional

Fuente: Grupo de Proyecto

Continuación Tabla 7: Funciones del Workflow

Ref.	Función	Categoría	Atributo	Detalles y Restricciones	Categoría
R1.2	Adjuntar a la solicitud las cotizaciones de los bienes necesarios.	Evidente	Metáfora de interfaz	(Det) Pantalla amigable (Det)Colorido	Obligatorio Opcional
			Facilidad de uso	(Res) máximo en 3 pasos deberá poder adjuntar doc.	Obligatorio
R1.3	Registrar estado de la solicitud	Evidente	Facilidad de uso	(Det) El usuario solo deberá hacer clic en una opción para realizar el cambio de estado de la solicitud.	Obligatorio
			Metáfora de interfaz	(Det) Pantalla amigable (Det)Colorido	Obligatorio Opcional
R1.4	Registrar ingreso de bienes adquiridos	Evidente	Metáfora de interfaz	(Det) Pantalla amigable (Det)Colorido	Obligatorio Opcional
R1.5	Enviar mails de avisos	Oculto	Tiempo de respuesta	(Res)máximo 3 segundos para que llegue mail	Obligatorio
R1.6	Generar reportes documentales	Evidente	Tiempo de respuesta	(Res)5 segundos como máximo	Obligatorio
R1.7	Generar Solicitud de Compra	Evidente	Tiempo de respuesta	(Res)5 segundos como máximo	Obligatorio
R1.8	Grabar	Evidente	Tiempo de respuesta	(Res)5 segundos como máximo	Obligatorio
R1.9	Imprimir	Evidente	Tiempo de respuesta	(Res)5 segundos como máximo	Obligatorio
R1.10	Validar Usuario	Oculto	Tiempo de respuesta	(Res)5 segundos como máximo	Obligatorio

Fuente: Grupo de Proyecto

## 2.8. Definiciones y Diagrama de Casos De Uso

### 2.8.1. Conceptos Básicos

**Casos de Uso.** Es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un workflow para completar un proceso.

**Actores.** Es una entidad externa del workflow que de alguna manera participa en la historia del caso de uso. Por lo regular estimula el workflow con eventos de entrada o recibe algo de el.

**Casos de uso de alto nivel.** Describe clara y concisamente un proceso, casi siempre en dos o tres enunciados.

**Casos de Usos Expandidos.** Consiste en destinar una parte para el curso normal de eventos donde se describe paso a paso un proceso.

**Escenario de un caso de uso.** Es una instancia o trayectoria realizada por medio del uso, un ejemplo real de su ejecución.

### 2.8.2 Definiciones de los Casos de Usos

#### Casos De Uso Expandidos

##### Primera Parte

**Caso de uso:** Crear petición

**Actores:** Secretaria, Director

**Tipo:** Primario

**Descripción:** La secretaria crea la petición de los bienes que se necesita adquirir para el Departamento, Escuela o Facultad y adjunta las cotizaciones correspondientes en caso de tenerlas.

**Referencias Cruzadas:** R1.1, R1.2, R1.5, R1.8, R1.10

### Curso normal de los Eventos

Tabla 8: Curso normal Crear Petición

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso comienza cuando la secretaria crea petición de bienes que se necesitan.	
2. La secretaria envía a revisión la petición.	3. Envío de mail al Rol que debe registrar la solicitud de petición.

Fuente: Grupo de Proyecto

### Cursos alternos:

Línea 2: La secretaria graba un borrador de la solicitud y no la envía enseguida.

**Caso de uso:** Registrar Petición

**Actores:** Director, Decano

**Tipo:** Primario

**Descripción:** El Director revisa la petición de los bienes que se necesita adquirir para el Departamento, Escuela o Facultad, y aprueba su envío al Decano.

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

## Curso normal de los Eventos

Tabla 9: Curso normal Registrar Petición

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso comienza cuando el Director lee mail que le informa que tiene una solicitud que registrar.	
2. El Director registra la petición y la envía a Decano.	3. Envío de mail al Rol que debe continuar tramitación.

Fuente: Grupo de Proyecto

### Cursos alternos:

Línea 2: Si la solicitud no está completamente bien la devuelve o rechaza.

**Caso de Uso:** Aprobar Solicitud

**Actores:** Decano, Administrador

**Tipo:** Primario

**Descripción:** El Decano inicia el trámite para la solicitud de petición.

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

## Curso normal de los Eventos

Tabla 10: Curso normal Aprobar Solicitud

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Decano lee mail que le indica que ha recibido una nueva solicitud de petición.	
2. Ingresa a la solicitud.	3. Muestra la solicitud de petición.
4. Realiza la revisión de la solicitud de petición y la aprueba.	5. Envío de mail al Rol que continúa el trámite

Fuente: Grupo de Proyecto

**Cursos alternos:**

Línea 4: El Decano rechaza la solicitud, por no haber sido aprobada por la Comisión Económica y el Consejo Directivo.

**Caso de Uso:** Realizar comparativo y generar Solicitud de Compra

**Actores:** Administrador, Jefe de Adquisiciones

**Tipo:** Primario

**Descripción:** El administrador revisa la solicitud de petición, realiza el cuadro comparativo de cotizaciones y adjunta éstas a la solicitud.

**Referencias Cruzadas:** R1.2, R1.3, R1.5, R1.8, R1.10

**Curso normal de los Eventos**

Tabla 11: Curso normal Realizar comparativo y generar Solicitud de Compra

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Administrador lee mail que le indica que tiene una solicitud por tramitar.	
2. Realiza la revisión de la solicitud de petición. Hace clic en botón Cuadro Comparativo.	3. Muestra una pantalla para el ingreso de datos de cotizaciones
4. El administrador ingresa los datos de los valores unitarios de los bienes solicitados y presiona botón Comparar.	5. Muestra los totales para cada empresa proveedora y también un mensaje con la empresa que tiene el precio menor.
6. El administrador abre opción para adjuntar cotizaciones.	7. Muestra la ventana donde el administrador selecciona los documentos que quiere adjuntar.
8. El administrador selecciona el documento y adjunta.	
9. El administrador regresa a la solicitud, genera la solicitud de compra y envía ambos documentos.	10. Envío de mail al Rol que continúa el trámite.

Fuente: Grupo de Proyecto

**Caso de Uso :** Registrar Verificación de proveedores

**Actores:** Vicerrector Administrativo, Jefe de Adquisiciones

**Tipo:** Primario

**Descripción:** El Vicerrector Administrativo verifica los proveedores de las cotizaciones adjuntas.

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

**Curso normal de los Eventos**

Tabla 12: Curso normal Registrar verificación de proveedores

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Vicerrector lee mail que indica que tiene una solicitud de petición por verificar.	
2. El Vicerrector extrae la documentación adjunta para su respectiva verificación.	3. Muestra la ventana para descargar los documentos.
4. El Vicerrector escoge entre abrir o guardar los documentos.	
5. El Vicerrector registra la verificación de los proveedores.	
6. El Vicerrector registra la verificación de los proveedores.	7. Envío de mail al Rol que continúa el trámite.

Fuente: Grupo de Proyecto

**Caso de Uso:** Registrar Verificación de cotizaciones.

**Actores:** Jefe de Adquisiciones, Jefe de Activo Fijo

**Tipo:** Primario

**Descripción:** El Jefe de Adquisiciones verifica las cotizaciones adjuntas a la solicitud de petición.

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

**Curso normal de los Eventos**

Tabla 13: Curso normal Registrar verificación de cotizaciones

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Jefe de adquisiciones lee mail que indica que tiene una solicitud de petición por verificar.	
2. El Jefe de adquisiciones extrae la documentación adjunta para su respectiva verificación.	3. Muestra la ventana para descargar los documentos.
4. El Jefe de adquisiciones escoge entre abrir o guardar los documentos.	
5. El Jefe de adquisiciones verifica las cotizaciones y aprueba la de menor valor.	6. Envío de mail al Rol que continúa el trámite.

Fuente: Grupo de Proyecto

## Segunda Parte

**Caso de Uso:** Registrar Despacho de bienes

**Actores:** Jefe de Activo Fijo, Administrador

**Tipo:** Primario

**Descripción:** El Jefe de Activo Fijo registra el despacho de los bienes adquiridos.

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

### Curso normal de los Eventos

Tabla 14: Curso normal Registrar despacho de bienes

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Jefe de Activo Fijo lee mail que indica que tiene que despachar los bienes adquiridos para una Facultad.	
2. El Jefe de Activo Fijo escoge la opción donde realiza el inventario de bienes.	3. Muestra una ventana para el ingreso de códigos.
4. El Jefe de Activo Fijo ingresa código para cada uno de los bienes que se van a despachar y registra el despacho de los mismos.	5. Envío de mail al Rol que continúa el trámite.

Fuente: Grupo de Proyecto

**Caso de Uso:** Registrar Recibido

**Actores:** Administrador, Bodeguero

**Tipo:** Primario

**Descripción:** El Administrador registrará un recibido en el momento que la solicitud tenga la aprobación de despacho (Activo Fijo).

**Referencias Cruzadas:** R1.3, R1.5, R1.8, R1.10

### **Curso normal de los Eventos**

Tabla 15: Curso normal Registrar recibido

<b>Acción de los Actores</b>	<b>Respuesta del Workflow</b>
1. Este caso de uso inicia cuando el administrador lee un mail que le indica que tiene que retirar los bienes.	
2. El administrador registra el recibido de los bienes.	3. Envío de mail al Rol que continúa el trámite.

Fuente: Grupo de Proyecto

**Caso de Uso:** Realizar Ingreso

**Actores:** Bodeguero, Director.

**Tipo:** Primario

**Descripción:** El bodeguero deberá realizar el ingreso de los bienes con su respectivo código, y comunicar al departamento solicitante el retiro de dichos bienes.

**Referencias Cruzadas:** R1.3, R1.4 , R1.5, R1.8, R1.10

## Curso normal de los Eventos

Tabla 16: Curso normal Realizar ingreso

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el bodeguero lee un mail que le indica que debe realizar el registro de los bienes recibidos.	
2. El bodeguero ingresa a opción para realizar el inventario interno de los bienes recibidos.	3. Muestra el documento con el formulario a llenar.
4. El bodeguero ingresa descripción y código de cada uno de los bienes. Envía la notificación al solicitante.	5. Envío de mail al Director para el retiro de los bienes.

Fuente: Grupo de Proyecto

**Caso de Uso:** Finalizar trámite

**Actores:** Director, Administrador

**Tipo:** Primario

**Descripción:** El Director registra el fin del trámite en el momento que los bienes tenga ya hecha la asignación (Bodega Facultad).

**Referencias Cruzadas:** R1.3 , R1.8, R1.10

## Curso normal de los Eventos

Tabla 17: Curso normal Finalizar trámite

Acción de los Actores	Respuesta del Workflow
1. Este caso de uso inicia cuando el Director lee un mail que le indica que los bienes solicitados ya están en bodega.	
2. El Director registra el fin del trámite.	3. Envío de mail al Administrador de finalización de trámite.

Fuente: Grupo de Proyecto

2.8.3 Diagrama de Caso de Uso

**CASOS DE USO PRIMERA PARTE PROCESO**

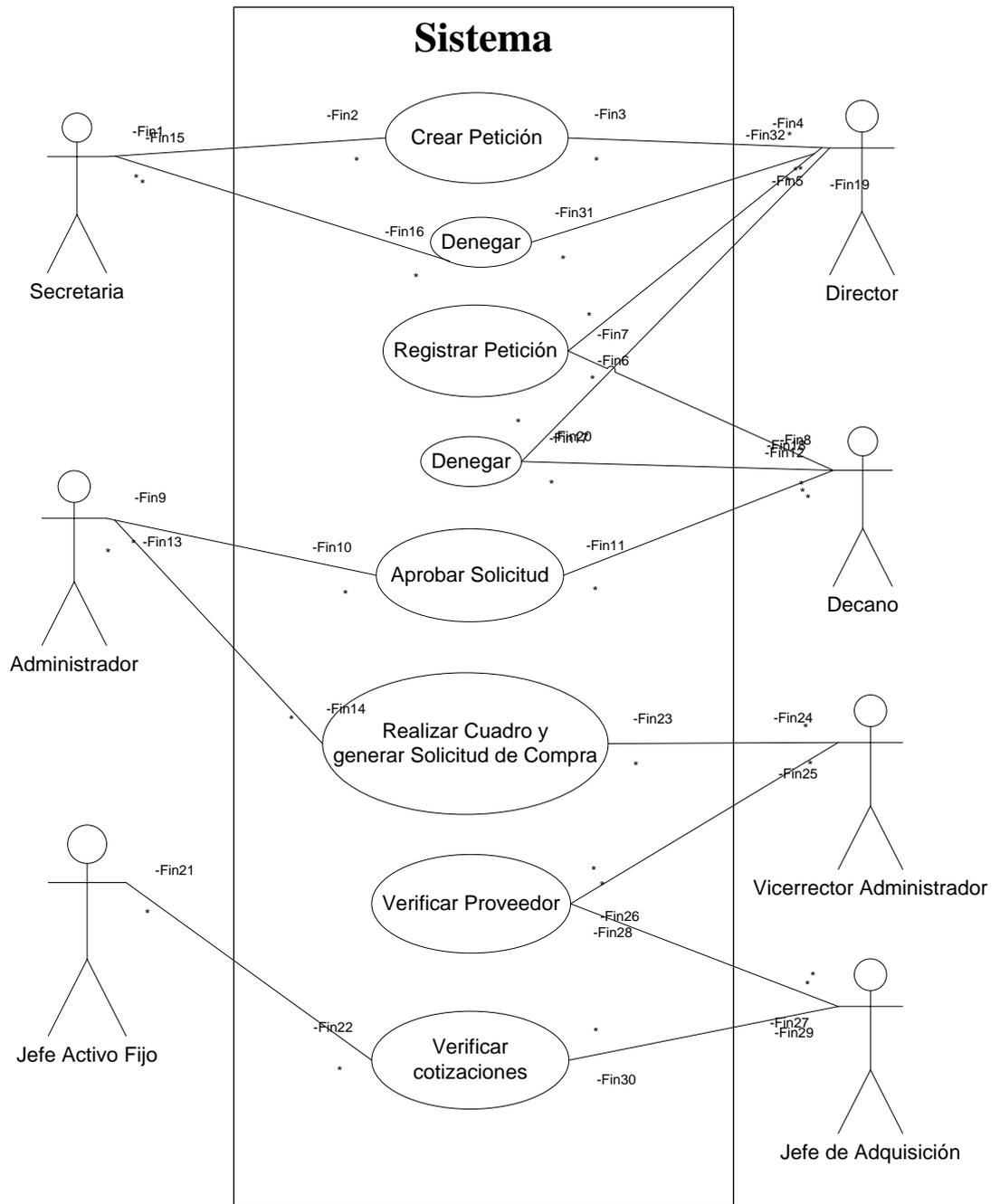


Figura 2.7

Fuente: Grupo de Proyecto

# CASOS DE USO SEGUNDA PARTE PROCESO

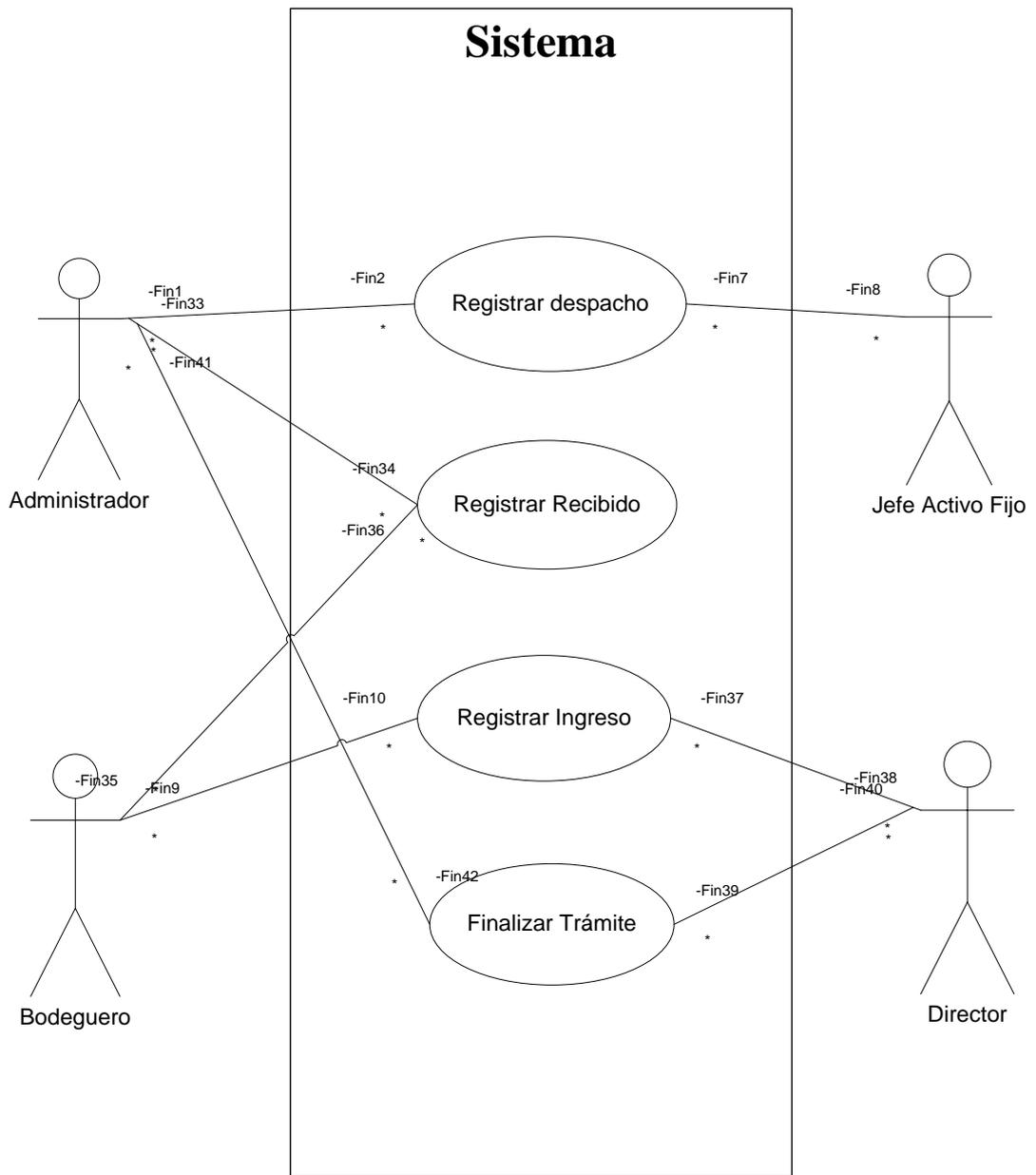


Figura 2.8

Fuente: Grupo de Proyecto

## 2.9 Modelo Conceptual

### 2.9.1 Conceptos Generales.

**Modelo Conceptual.** Es una representación de conceptos en un dominio del problema real, no es una descripción del diseño de software.

El modelo conceptual puede mostrar:

- Conceptos
- Asociaciones entre conceptos
- Atributos de conceptos

**Conceptos.** Es una idea, cosa u objeto. La creación de un modelo conceptual se comienza preparando una lista de conceptos idóneos.

### Representación del Concepto

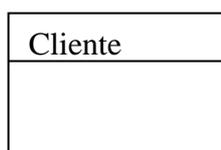


Figura 2.9

Fuente: Grupo de Proyecto

**Asociaciones.** Es una relación entre dos conceptos que indica alguna conexión significativa e interesante entre ellos.

**Multiplicidad.** Define cuantas instancias de Tipo A se pueden asociar con una instancia de tipo B

### Representación de Multiplicidad

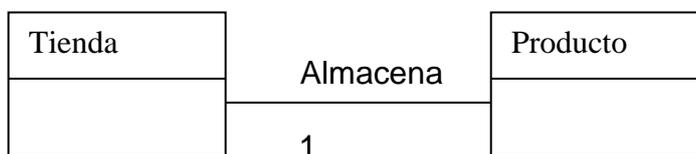


Figura 2.10

Fuente: Grupo de Proyecto

**Atributos.** Es un valor lógico de un dato de un objeto.

### Representación de Atributo

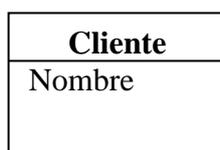


Figura 2.11

Fuente: Grupo de Proyecto

## 2.10. Diagrama De Secuencia

**Diagrama de Secuencia.** Es una representación que muestra, el curso particular de los eventos de un caso de uso, los actores externos que interactúan directamente con el workflow, y los eventos del workflow generados por los actores.

**Evento de un workflow.** Es un hecho externo de entrada que un actor produce en un workflow.

**Operación de un workflow** es una acción que éste ejecuta en respuesta a un evento del workflow.

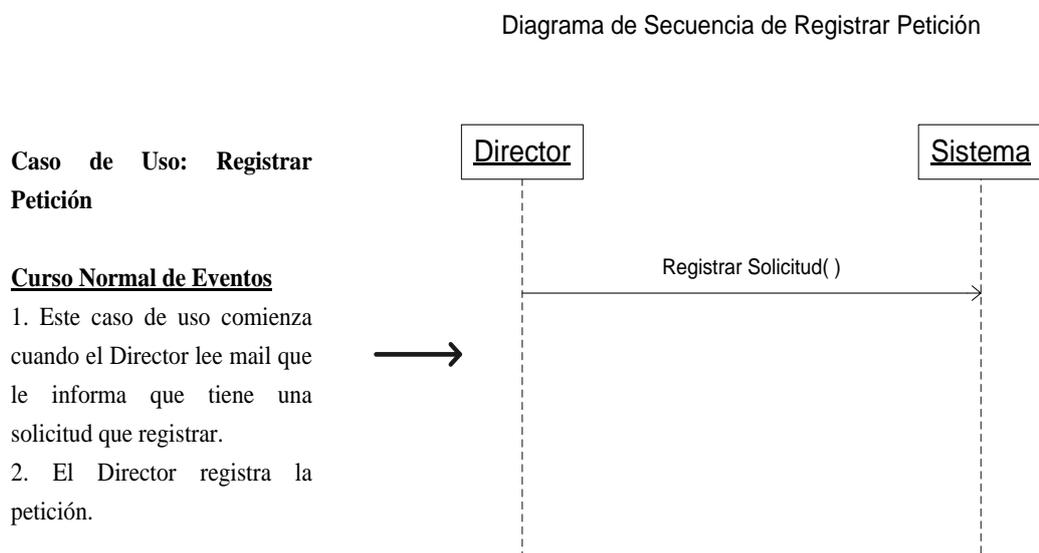
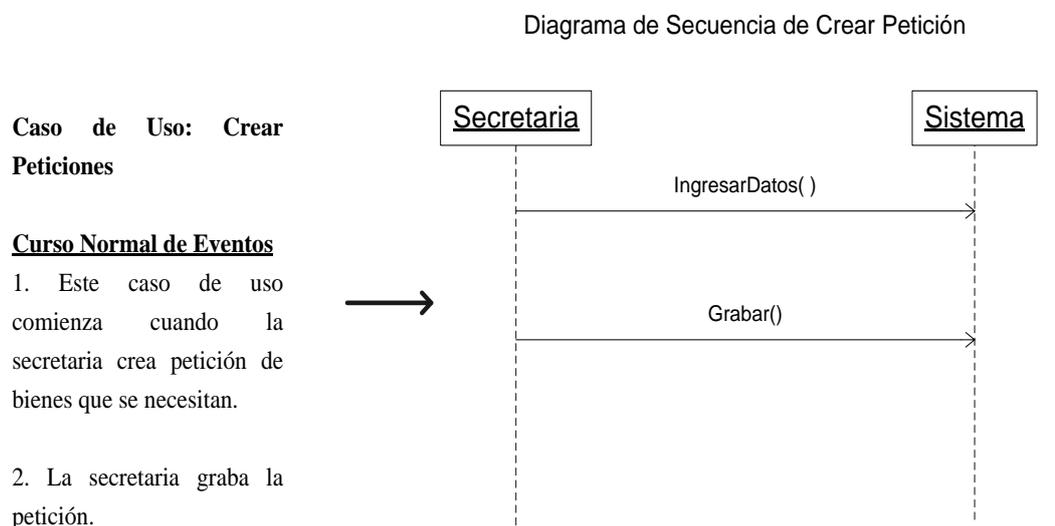


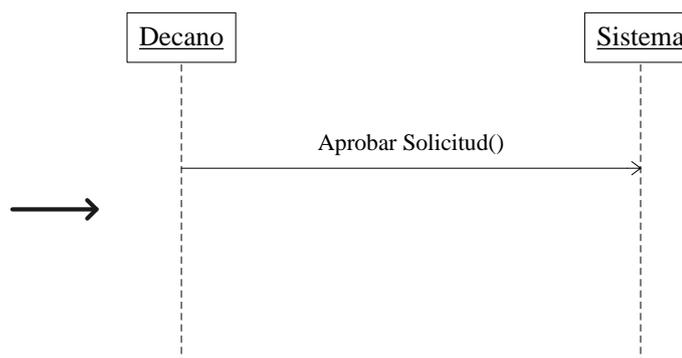
Figura 2.12

Fuente: Grupo de Proyecto

**Caso de Uso: Aprobar Solicitud**

**Curso Normal de Evento**

1. Este caso de uso inicia cuando el Decano lee mail que le indica que ha recibido una nueva solicitud de petición.
2. Realiza la revisión de la solicitud de petición.
3. El Decano aprueba el trámite.



**Caso de Uso: Realizar Comparativo y Generar Solicitud de Compra**

**Curso normal de Evento**

1. Este caso de uso inicia cuando el Administrador lee mail que le indica que tiene una solicitud por tramitar.
2. Realiza la revisión de la solicitud de petición.
3. El administrador realiza el cuadro comparativo de las cotizaciones y las adjunta .
4. El administrador graba los cambios.
5. El administrador envía la documentación.

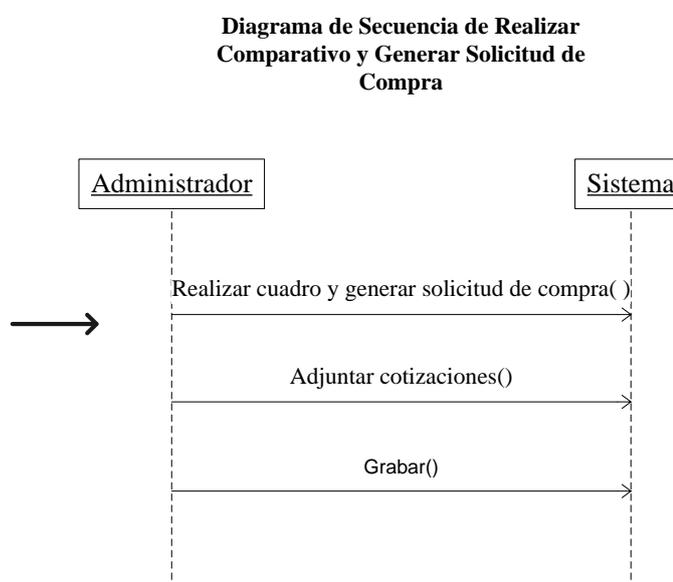


Figura 2.13

Fuente: Grupo de Proyecto

### Diagrama de Secuencia de Verificar Proveedor

#### Caso de Uso: Verificar Proveedor

##### Curso Normal de Eventos

1. Este caso de uso inicia cuando el Vicerrector lee mail que indica que tiene una solicitud de petición por verificar.
2. El Vicerrector extrae la documentación adjunta para su respectiva verificación.
4. El Vicerrector escoge entre abrir o guardar los documentos.
5. El Vicerrector registra la verificación de los proveedores.



### Diagrama de Secuencia de Verificar Cotizaciones

#### Caso de Uso: Verificar Cotizaciones

##### Curso Normal de Eventos

1. Este caso de uso inicia cuando el Jefe de Adquisición lee mail que indica que tiene una solicitud de petición por verificar.
2. El Jefe de Adquisición extrae la documentación adjunta para su respectiva verificación.
4. El Jefe de Adquisición escoge entre abrir o guardar los documentos.
5. El Jefe de Adquisición registra la verificación de los cotizaciones.

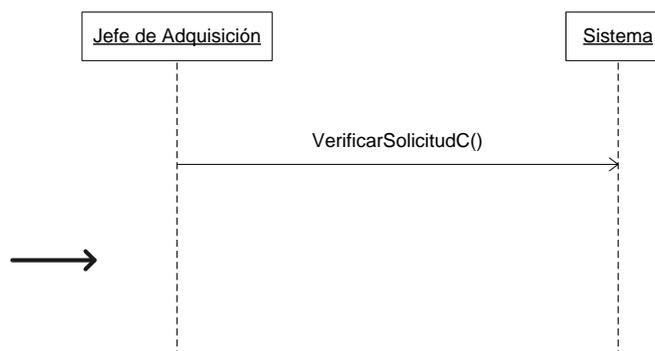


Figura 2.14

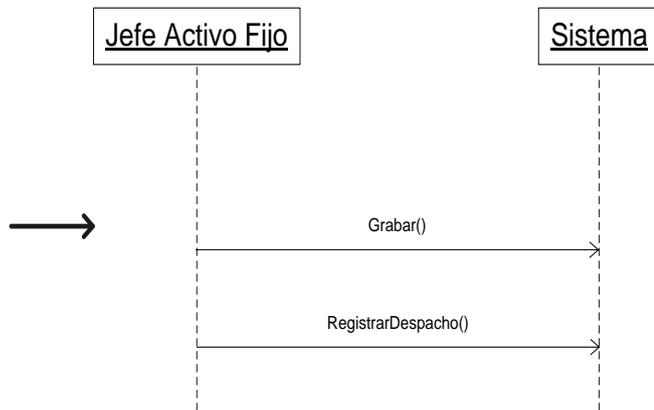
Fuente: Grupo de Proyecto

### Diagrama de Secuencia de Registrar Despacho

#### Caso de Uso: Registrar Despacho

##### Curso Normal de Eventos

1. Este caso de uso inicia cuando el Jefe de Activo Fijo lee mail que indica que tiene que despachar los bienes adquiridos para una Facultad.
2. El Jefe de Adquisiciones codifica los bienes adquiridos.
4. El Jefe de Adquisiciones graba la codificación de los bienes.
5. El Jefe de Adquisiciones registra el despacho de los bienes.



### Diagrama de Secuencia de Registrar Recibido

#### Caso de Uso: Registrar Recibido

##### Curso Normal de Eventos

1. Este caso de uso inicia cuando el administrador lee un mail que le indica que tiene que retirar los bienes.
2. El administrador registra el recibido de los bienes.

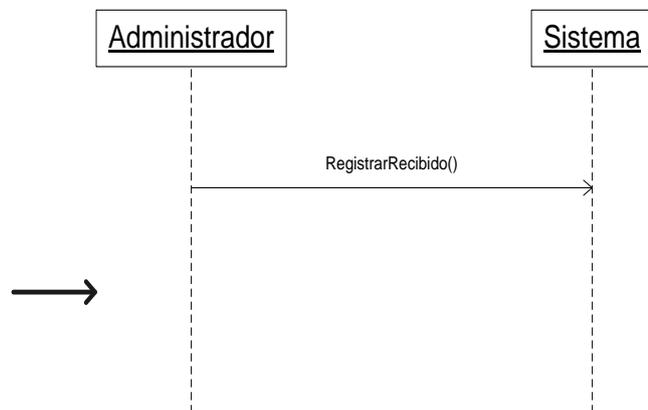


Figura 2.15

Fuente: Grupo de Proyecto

Diagrama de Secuencia de Realizar Ingreso

**Caso de Uso: Realizar Ingreso**

**Curso Normal de Eventos**

1. Este caso de uso inicia cuando el bodeguero lee un mail que le indica que debe realizar el registro de los bienes recibidos.
2. El bodeguero realiza el inventario interno de los bienes recibidos.
4. El bodeguero ingresa los datos de los bienes.
5. Graba los datos

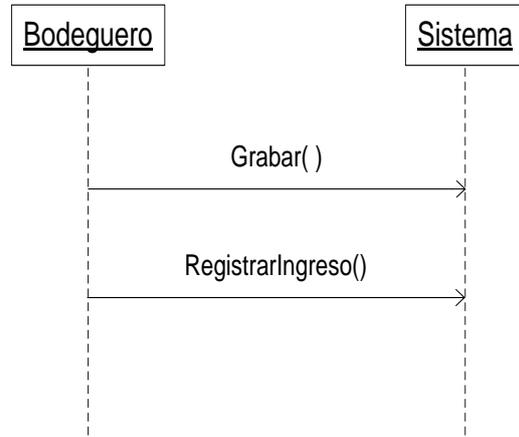


Diagrama de Secuencia de Finalizar Trámite

**Caso de Uso: Finalizar Trámite**

**Curso Normal de Evento**

1. Este caso de uso inicia cuando el Director lee un mail que le indica que tiene que retirar los bienes.
2. El Director registra el fin de trámite.

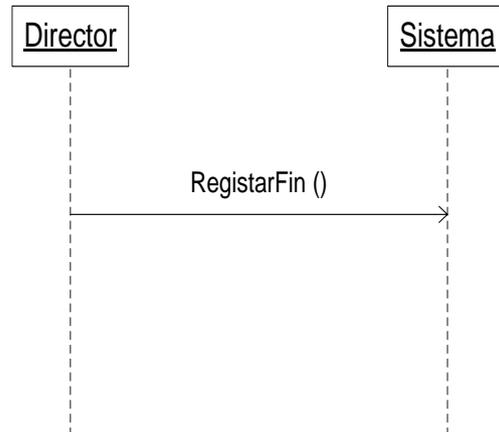


Figura 2.16

Fuente: Grupo de Proyecto

## 2.11. Contratos

**Contrato.** Es un documento que describe lo que una operación se propone lograr.

**Contrato de operación del workflow.** Describe los cambios del estado del workflow total cuando se llama a una de sus operaciones.

### Secciones de un Contrato

<b>Nombre:</b>	Nombre de la operación y parámetros
<b>Responsabilidades:</b>	Descripción informal de las responsabilidades que debe cumplir la operación
<b>Tipo:</b>	Nombre del Tipo (concepto, clase de software, interfaz)
<b>Referencias Cruzadas:</b>	Números de las funciones del workflow, casos uso
<b>Notas:</b>	Notas de diseño, algoritmos e información a fin
<b>Excepciones:</b>	Casos excepcionales
<b>Precondiciones:</b>	Suposiciones acerca del estado del workflow antes de ejecuta la operación
<b>Poscondiciones:</b>	El estado del workflow después de la operación.

## **Contratos para el caso de uso Crear Peticiones**

### **Contrato para Crear Solicitud**

**Nombre:** IngresarDatos()

**Responsabilidades:** Registrar datos de los bienes

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.1, R1.5

**Notas:**

**Excepciones:**

**Precondiciones:**

**Poscondiciones:**

Si se trata de una nueva solicitud de peticiones. Una Solicitud fue creada (creación de instancia)

Si se trata de una nueva solicitud, la Solicitud fue asociada a Bien (asociación formada)

Si se trata de una nueva solicitud de peticiones. Un Mail es enviado, instancia fue creada (creación de instancia)

Si se trata de una nueva solicitud, el Mail es asociado a Director (asociación formada)

**Contrato para Grabar**

**Nombre:** Grabar()  
**Responsabilidades:** Grabar los datos en la Solicitud.  
**Tipo:** Workflow  
**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.8  
**Notas:**

**Excepciones:**

**Precondiciones:** La información debe estar completa

**Poscondiciones:**

Si se trata de una nueva solicitud de peticiones. Un solicitud es creada, instancia fue creada (creación de instancia)

Si se trata de una nueva solicitud, la solicitud es asociada a Secretariat (asociación formada)

**Contrato para el caso de uso Registrar Peticiones**

**Nombre:** RegistrarSolicitud()  
**Responsabilidades:** Cambio de estado de la Petición envío de mail.  
**Tipo:** Workflow  
**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.8, R1.5  
**Notas:**  
**Excepciones:** No registrar la solicitud

**Precondiciones:****Poscondiciones:**

Se asoció Solicitud a Director

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail al Decano (asociación formada).

**Contrato para el caso de uso Aprobar Solicitud**

**Nombre:** Aprobar()

**Responsabilidades:** Cambio de estado de la Petición, envío de mail

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.8, R1.5

**Notas:**

**Excepciones:** Que la solicitud sea rechazada

**Precondiciones:** El estado se encuentre en registrado

**Poscondiciones:**

Se asoció Solicitud a Decano

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail al Administrador (asociación formada).

**Contratos para el caso de uso Realizar Comparativo y Generar Solicitud de Compra**

**Contrato para Realizar Comparativo y Generar Solicitud de Compra**

**Nombre:** Comparativo y Genera Solicitud Compra ()

**Responsabilidades:** Realizar cuadro comparativo de cotizaciones, Cambio de estado de la Petición, Generar solicitud de compra, envío de mail.

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.2, R1.5, R1.7

**Notas:**

**Excepciones:**

**Precondiciones:** Estar en estado de aprobado

**Poscondiciones:**

Se asoció Solicitud a Administrador

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Solicitud Compra, instancia fue creada (creación de instancia)

Se asignó Estado.Solicitud Compra (Modificación atributo)

Se asoció instancia Solicitud Compra a Solicitud (asociación formada).

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Vicerrector Administrativo (asociación formada).

### **Contrato para Grabar**

**Nombre:** Grabar()

**Responsabilidades:** Grabar los datos en la Solicitud.

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.8

**Notas:**

**Excepciones:**

**Precondiciones:** La información debe estar completa

**Poscondiciones:**

Si se trata de un nuevo cuadro comparativo. Un cuadro es creado, instancia fue creada (creación de instancia)

Si se trata de un nuevo cuadro comparativo, el cuadro es asociado a Administrador (asociación formada)

### **Contrato para el caso de uso Verificar Proveedor**

**Nombre:** VerificarSolicitud()

**Responsabilidades:** Cambio de estado en la solicitud, envío de mail

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.5, R1.8

**Notas:**

**Excepciones:**

**Precondiciones:** Toda la documentación completa, estado de la petición en aprobada

**Poscondiciones:**

Se asoció instancia Solicitud a Vicerrector (asociación formada).

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Jefe Adquisición (asociación formada).

### **Contrato para el caso de uso Verificar Cotizaciones**

**Nombre:** VerificarSolicitudC()

**Responsabilidades:** Cambio de estado en la solicitud, envío de mail.

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.5, R1.8

**Notas:** Al enviar esta solicitud se irá al módulo de compras.

**Excepciones:**

**Precondiciones:** Que las solicitudes tengan adjuntas cotizaciones

**Poscondiciones:**

Se asoció instancia Solicitud a Jefe de Adquisición (asociación formada).

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

**Contratos para el caso de uso Registrar Despacho****Contrato para el caso de uso Registrar Despacho**

**Nombre:** RegistrarEstado()

**Responsabilidades:** Cambio de estado de la solicitud, envío de mail

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.5, R1.4

**Notas:** La codificación de los bienes será usada en el inventario interno de la facultad.

**Excepciones:**

**Precondiciones:** Que el departamento de Proveduría comunique al Jefe de Activo fijo que los bienes han sido adquiridos.

**Poscondiciones:**

Se asoció instancia Solicitud a Jefe de Activo Fijo (asociación formada).

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Administrador (asociación formada).

**Contrato para Grabar**

**Nombre:** Grabar()

**Responsabilidades:** Grabar la codificación de los bienes.

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.8

**Notas:**

**Excepciones:**

**Precondiciones:** La información debe estar completa

**Poscondiciones:**

Se asoció atributo código a instancia Bien (asociación formada).

**Contrato para el caso de uso Registrar Recibido**

**Nombre:** RegistrarRecibido()

**Responsabilidades:** Cambio de estado en el oficio de codificación de bienes, envío de mail

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.5, R1.8

**Notas:**

**Excepciones:**

**Precondiciones:**

**Poscondiciones:**

Se asoció instancia Solicitud a Administrador (asociación formada).

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Bodeguero (asociación formada).

### **Contratos para el caso de uso Realizar Ingreso**

#### **Contrato para el caso de uso Realizar Ingreso**

**Nombre:** RealizarIngreso()

**Responsabilidades:** Cambio de estado en el oficio de  
codificación de bienes,  
Ingreso de los códigos de los bienes

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.4, R1.8, R1.5,  
R1.6

**Notas:**

**Excepciones:**

**Precondiciones:** Que los bienes hayan sido retirados por el Administrador

**Poscondiciones:**

Se asoció instancia Solicitud a Bodeguero (asociación formada).

Se asignó Estado.Solicitud (Modificación atributo)

Se asoció instancia Bodeguero a Bien (asociación formada)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Director (asociación formada).

**Contrato para Grabar**

**Nombre:** Grabar()

**Responsabilidades:** Grabar la codificación de los bienes más su asignación y responsable.

**Tipo:** Workflow

**Referencias Cruzadas:** Funciones de workflow: R1.8

**Notas:**

**Excepciones:**

**Precondiciones:** La información debe estar completa

**Poscondiciones:**

Se asoció atributo responsable a instancia Bien (asociación formada).

**Contrato para el caso de uso Finalizar Trámite****Contrato para Finalizar Trámite****Nombre:** RegistrarFin()**Responsabilidades:** Cambio de estado en la solicitud, envío de mail**Tipo:** Workflow**Referencias Cruzadas:** Funciones de workflow: R1.3, R1.8, R1.5**Notas:****Excepciones:****Precondiciones:** Recibir mail de confirmación de bienes ingresados a bodega**Poscondiciones:**

Se asignó Estado.Solicitud (Modificación atributo)

Se crea Mail, instancia fue creada (creación de instancia)

Se asoció instancia Mail a Administrador (asociación formada).

# CAPITULO 3

## 3. DISEÑO Y DESARROLLO

La herramienta Lotus Domino maneja el diseño a través del Lotus Domino Designer, el cual crea aplicaciones de Domino.

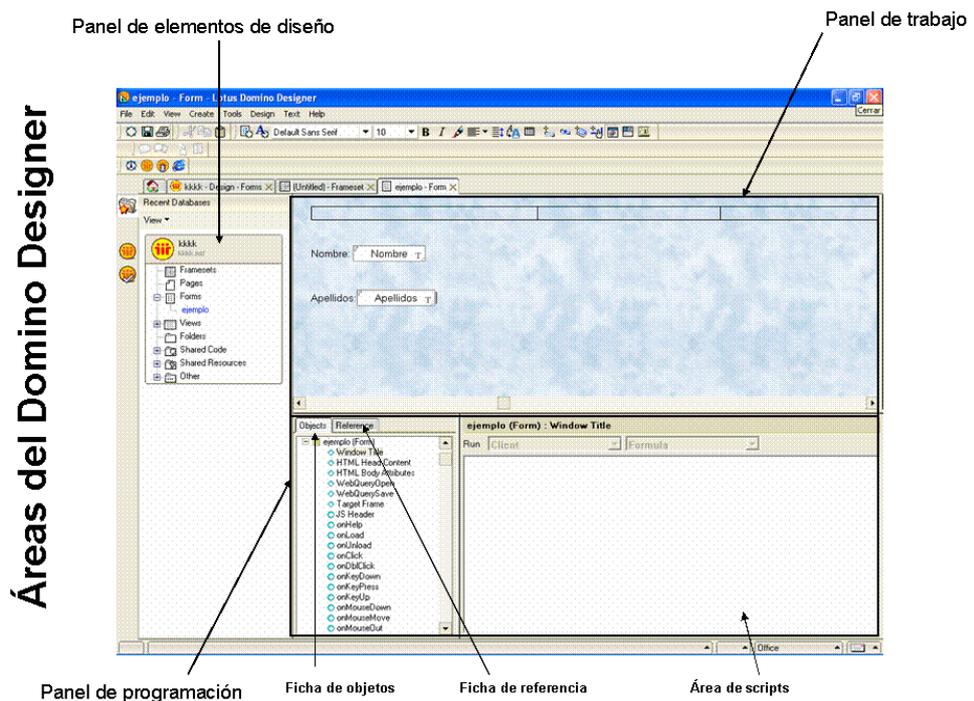


Figura 3.1

Fuente: Grupo de Proyecto

**Panel de diseño:** Incluye el **panel de elementos de diseño** para la base de datos. Esta es la lista de los elementos de diseño y de los **iconos de los marcadores de diseño**, que permiten acceder a las últimas bases de datos con las que ha trabajado.

**Panel de trabajo:** Se utiliza para crear y modificar formularios, **páginas**, vistas y otros elementos del panel de elementos de diseño específicos de su aplicación o base de datos. Cuando se abre o se crea un elemento de diseño individual, Domino Designer muestra automáticamente dicho elemento en el panel de trabajo.

**Panel de programación:** Se utiliza para agregar código a determinadas áreas del elemento de diseño actual que aparece en el panel de trabajo. Para el Workflow se utilizó Lenguaje de Fórmulas, Lotus Script, Java Script.

### **3.1 Elementos de diseño**

Son las piezas con las que se construyen las aplicaciones. Puede personalizar estas estructuras para crear aplicaciones.

### 3.1.1 Tareas de las aplicaciones y los elementos de diseño correspondientes

Tabla 18: Elementos de diseño

Tarea o funcionalidad de la aplicación	Elementos de diseño
Mostrar, recopilar y organizar información	<ul style="list-style-type: none"> <li>• <b>Páginas:</b> Contienen texto, imágenes, <i>applets</i> y otros objetos. Los usuarios no pueden introducir información en estos elementos.</li> <li>• <b>Formularios:</b> Contienen campos en los que los usuarios pueden introducir valores. Estos valores se almacenan como un <b>documento</b>.</li> <li>• <b>Vistas:</b> Ordenan y muestran listas de documentos incluidos en la aplicación.</li> </ul>
Automatizar tareas	<b>Acciones y agentes:</b> Pueden ejecutar <i>scripts</i> cuando un usuario realiza alguna acción.
Controlar la navegación	<b>Zonas activas, conjuntos de marcos, esquemas y mapas de imágenes:</b> Su uso combinado permite a los usuarios desplazarse rápida y fácilmente por la aplicación.

Fuente: Manual de Lotus Domino

### 3.1.2 Acceso a los elementos de diseño

Cuando se hace clic en un elemento de diseño en el panel de elementos de diseño, todos los elementos individuales específicos de la base de datos que pertenecen a ese tipo de elemento aparecen en el panel de diseño y en el panel de trabajo. Para abrir un elemento de diseño, se debe dar doble clic en él, en el panel de trabajo. El elemento se abrirá en este panel, con el panel de programación debajo.

### 3.1.3 Uso de formularios

Un formulario es un elemento de diseño que permite introducir y mostrar datos. Los formularios contienen la lógica de aplicación y pueden realizar acciones sobre los datos. Los datos que se recopilan a través de un formulario se almacenan en una nota de datos, también denominada documento.

Cuando un usuario agrega información a una base de datos usando un formulario y lo guarda, dicha información se guarda como un documento independiente del formulario. Cuando un usuario abre un documento, este utiliza el formulario como plantilla que le proporciona la estructura necesaria para mostrar los datos.

Los formularios pueden utilizar los siguientes elementos:

Tabla 19: Uso de formularios

Elemento	Función
Elementos estáticos (Texto, gráficos)	Muestran información o gráficos en todos los documentos que utilizan el formulario.
Campos	<ul style="list-style-type: none"> <li>✓ Muestran la información de los documentos.</li> <li>✓ Guardan la información en un documento.</li> <li>✓ Calculan un valor.</li> </ul>
Propiedades	Controlan los atributos del formulario. Por ejemplo, el color de fondo es una propiedad de formulario.
Otros elementos, como por ejemplo: <ul style="list-style-type: none"> <li>◆ Subformularios</li> <li>◆ Campos compartidos</li> <li>◆ Botones y acciones (Lógica de programación)</li> </ul>	Ofrecen al diseñador herramientas adicionales que suelen mejorar la interactividad o la eficacia del formulario.

Fuente: Manual de Lotus Domino

La figura siguiente representa el elemento de diseño formulario.

### Elementos de Diseño de un Formulario

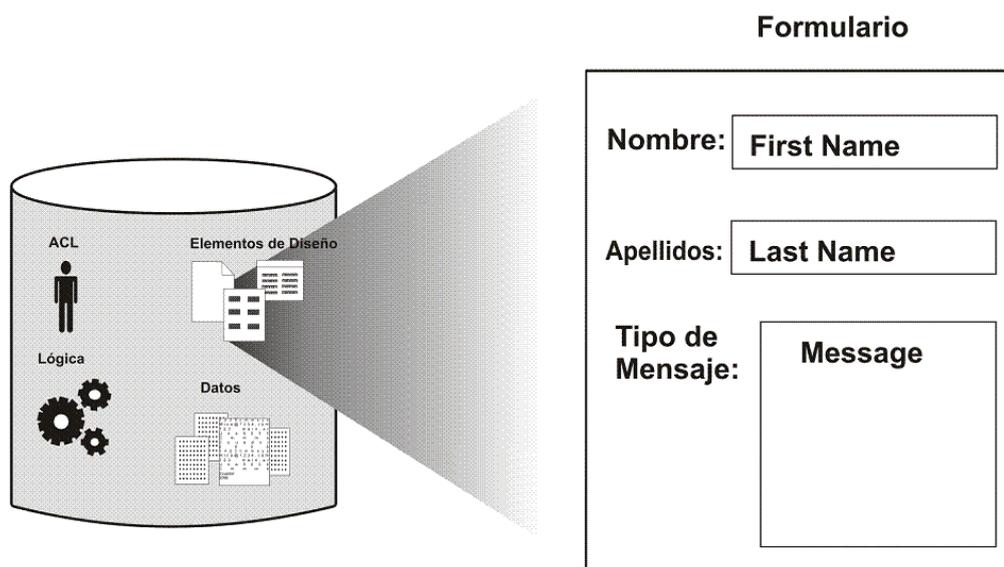


Figura 3.2

Fuente: Manual de Lotus Domino

### 3.1.4 Notas de bases de datos

Una nota no es más que un documento, el cual guarda información.

Las bases de datos de Domino contienen cuatro tipos distintos de notas.

Tabla 20: Tipos de Notas

Tipo de nota	Función
Encabezado	Contiene información sobre la propia base de datos, como por ejemplo, su nombre, tamaño y fecha de creación. Cada base de datos tiene una única nota de encabezado.
Nota de LCA	Guarda la Lista de Control de Acceso, una lista de nombres y permisos asociados a la base de datos.
Nota de diseño	Cada elemento de diseño, como por ejemplo, un formulario, una vista o una página, se almacena en la base de datos como una nota de diseño. La nota de diseño de formulario actúa como plantilla para crear nuevos documentos o mostrar información.
Nota de datos	Una nota de datos contiene la información introducida a través de un formulario, así como los datos del workflow utilizados por Domino para las labores de gestión.

Fuente: Manual de Lotus Domino

### 3.1.5 Uso de las vistas

La finalidad del elemento de diseño vista es crear una lista organizada de documentos para que los usuarios puedan localizar la información que necesitan.

Las vistas son uno de los puntos de entrada a los documentos que se encuentran almacenados en una base de datos de Domino. Todas las bases de datos contienen al menos una vista. Sin embargo, la mayor parte de ellas contienen varias vistas.

El elemento de diseño vista esta representado en el gráfico a continuación.

## Elementos de Diseño Vista

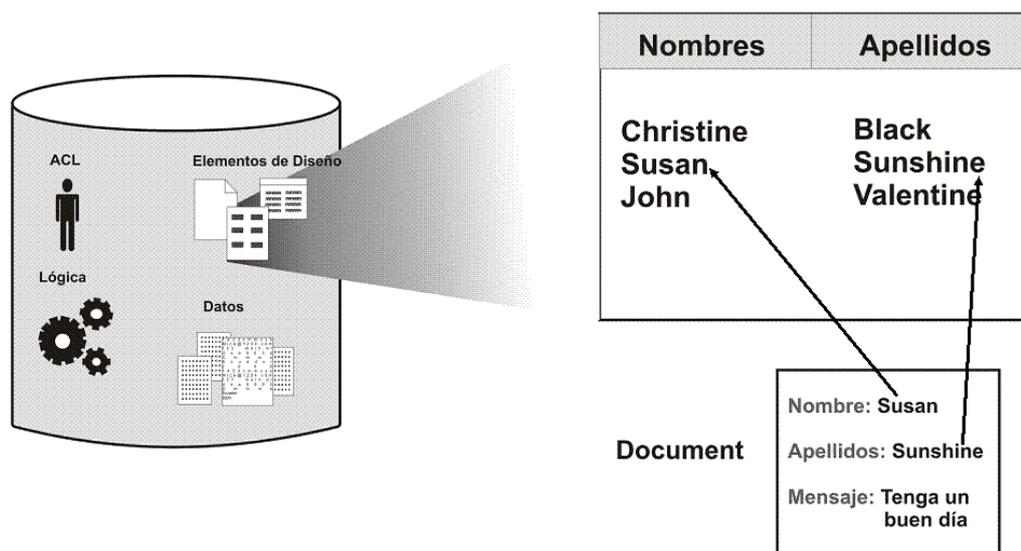


Figura 3.3

Fuente: Manual de Lotus Domino

### 3.1.5.1 Características de una vista

Algunas de las características de las vistas son:

- Una vista es una lista de documentos de una base de datos.
- Cada fila de la vista representa un documento de la base de datos.
- Cada columna muestra información sobre el documento de esa fila.
- Una vista puede incluir todos los documentos o un subconjunto de ellos basándose en criterios de selección.
- Los usuarios leen la información de la vista para localizar documentos específicos.

- Un número excesivo de vistas, así como un diseño incorrecto de estas, pueden afectar negativamente al rendimiento de la base de datos.
- El diseño de las vistas debería realizarse en las primeras etapas del proceso de desarrollo de las aplicaciones.

### 3.1.5.2 Almacenamiento de las vistas

Domino utiliza un workflow interno de localización de documentos, denominado índice de vista, para almacenar la lista de documentos de una vista. Los usuarios finales y los diseñadores nunca trabajan directamente con el índice de vista. La aplicación de Domino es la encargada de crearlo y mantenerlo.

### 3.1.5.3 Creación de una vista

Tabla 21: Creación de una vista

Fase	Descripción
1	El desarrollador de aplicaciones crea el diseño de la vista.
2	Domino extrae información de los documentos de acuerdo con el diseño y crea un índice de vista.
3	Domino muestra la información a los usuarios.

Fuente: Manual de Lotus Domino

## Cómo crea Domino las vistas

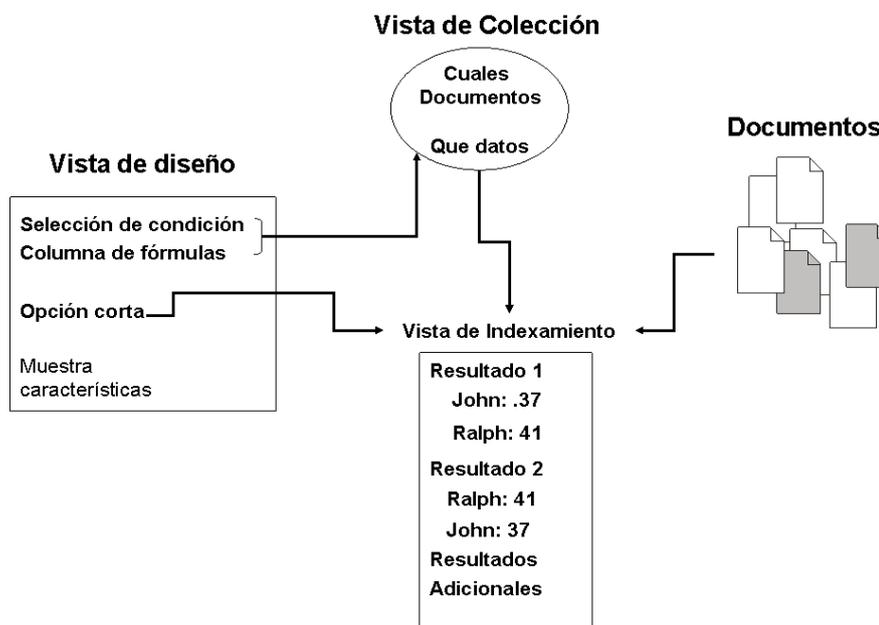


Figura 3.4

Fuente: Manual de Lotus Domino

### 3.1.5.4 Facilidad de Indexamiento Notes

La Facilidad de Indexamiento Notes reside en las bases de datos de Domino. Esta utilidad es responsable de todas las tareas que mantienen ordenados los documentos en cada una de las vistas de la base de datos. Entre sus funciones se encuentran:

- La apertura y el cierre de conjuntos de datos de vista
- La actualización del índice
- La localización de notas en un índice

- La localización de entradas en un índice
- La actualización de conjuntos de datos de vista

### **3.1.6 Uso de los recursos compartidos**

Un recurso compartido es un elemento, como por ejemplo, un archivo gráfico, almacenado en una base de datos de Domino. Cada base de datos puede contener una biblioteca con sus propios recursos compartidos. Los recursos compartidos le permiten hacer referencia a un recurso varias veces en una base de datos, pero manteniendo dicho recurso en una única ubicación.

Para poder utilizar un recurso compartido, es necesario importarlo en la base de datos de Domino. Su formato original se conserva durante este proceso.

#### **3.1.6.1 Tipos de recursos compartidos**

Cada base de datos puede contener una biblioteca de recursos compartidos con:

- Imágenes
- Archivos
- Applets

- Hojas de estilo
- Conexiones de datos

### **3.1.6.2 Ventajas de los recursos compartidos**

El uso de los recursos compartidos reduce el mantenimiento de la base de datos. Si se modifica un recurso, todas las referencias a dicho recurso se actualizan automáticamente. Una gran ventaja de los recursos compartidos es que forman parte de la aplicación; si esta se traslada, se copia o se replica, también lo harán los recursos compartidos.

### **3.1.7 Formularios relacionados**

En la mayoría de las aplicaciones existe la necesidad de crear información que esta relacionada con otra información ya existente.

En la realización de este proyecto, se hará uso de esta opción que brinda Lotus Domino, para la creación de la Solicitud de Compra que debe generarse a partir de la Solicitud de Petición, al momento de ser aprobada por el Decano de la Facultad.

### 3.1.7.1 Tipos de formularios

Lotus Domino dispone de varios tipos de formularios que permiten al desarrollador utilizar este tipo de estructura de pregunta/respuesta, denominada generalmente jerarquía de respuestas, los cuales se detallan a continuación:

- Documento: este tipo de documento es el tipo predeterminado para los formularios de Domino. Se utiliza para crear el nivel superior en una jerarquía de documentos. Estos documentos se denominan generalmente documentos principales.
- Respuesta: crea documentos de respuesta asociados con un documento principal. Los documentos de respuesta también se denominan documentos de nivel inferior.
- Respuesta a respuesta: crea documentos de respuesta asociados con un documento principal o con otro documento de respuesta.

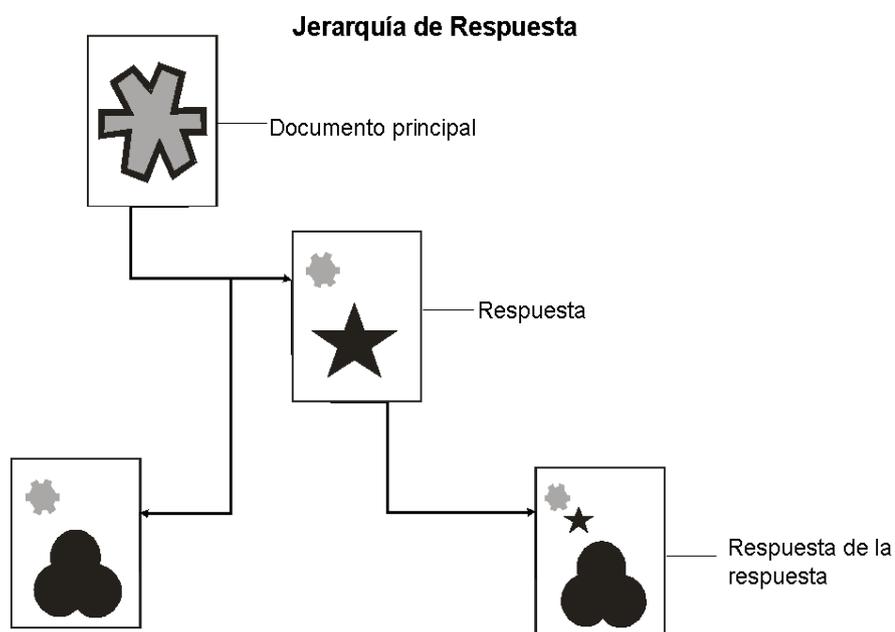


Figura 3.5

Fuente: Manual de Lotus Domino

De forma predeterminada, el tipo de formulario de Domino es Documento. Para designar un formulario como formulario de Respuesta o de Respuesta a Respuesta, se debe cambiar la propiedad Tipo del formulario.

Para establecer correctamente la relación entre un documento principal y un documento de respuesta, al crear la respuesta es necesario tener abierto el documento principal o tenerlo seleccionado en una vista.

Mediante las propiedades del formulario de Domino, el desarrollador puede diseñar el formulario de respuesta de forma que:

- Se cree un vínculo al documento principal en un campo de texto enriquecido.
- Se copie el contenido del documento principal como texto enriquecido plegable.
- Se copie el contenido del documento principal en un campo de texto enriquecido.

La creación de un vínculo al documento principal consume mucho menos espacio en disco y es el método más habitual.

### **3.1.8 Uso de acciones**

Una acción es un programa asociado con una vista o un formulario que se emplea para automatizar las tareas de los usuarios. Las acciones se agregan a las aplicaciones para:

- Acelerar las tareas repetitivas
- Actualizar la información
- Simplificar el uso de la base de datos
- Realizar cálculos complejos
- Comprobar la existencia de errores.

## Ejemplos de acciones

Una acción puede emplearse para:

- Recrear los comandos de los menús de Notes como botones. Por ejemplo:
  - ◆ Crear un documento
  - ◆ Editar un documento
  - ◆ Guardar un documento
- Enviar por correo el documento actual
- Calcular formulas complejas que no necesitan guardarse con los documentos individuales.

### 3.1.8.1 Las acciones y los usuarios

Las acciones de una vista están a disposición de los usuarios cuando se abre la vista. Las acciones de un formulario están disponibles cuando se visualiza un documento creado con dicho formulario.

Los usuarios acceden a las acciones desde:

- El menú Acciones de Notes
- La barra de acciones

La barra de acciones es una fila de botones situada en la parte superior de una vista o un formulario. Para ejecutar una acción de la barra de acciones, los usuarios solo tienen que dar clic en el botón.

### **3.1.9 Uso de los agentes**

Los agentes son programas de una aplicación de Domino. Entre las ventajas del uso de agentes se encuentran las siguientes:

- Los agentes pueden ejecutarse manualmente los usuarios o pueden ejecutarse automáticamente en segundo plano como agentes programados.
- Los agentes no están asociados con ningún elemento de diseño específico.
- Los agentes se pueden ejecutar en un servidor específico, en varios servidores, en estaciones de trabajo o en web
- Los agentes pueden ejecutar otros agentes
- Los agentes pueden estar compuestos por acciones simples, fórmulas, scripts de LotusScript o programas de Java
- Los agentes se pueden distribuir fácilmente porque pueden replicarse
- Los agentes pueden ser compartidos o privados

### **3.1.9.1 Acciones frente a agentes**

Los agentes se diferencian de las acciones en lo siguiente:

- Las acciones de formulario o de vista solo funcionan cuando el formulario o la vista están abiertos
- Los agentes no están limitados a un formulario o una vista determinados. Se los puede usar en cualquier lugar de la aplicación de Domino para configurar tareas activadas por el usuario o que se ejecutaran en segundo plano.

En la elaboración de este proyecto se utilizará el agente para el envío del mail a las diferentes instancias.

## **3.2. Generalidades sobre Base de Datos**

Lotus Domino utiliza bases de datos documentales, es decir una base de datos contendrá documentos a través de los cuales se accede a la información deseada, a diferencia de las bases de datos relacionales, una base de datos documental no contiene tablas con campos por los que se accesa a la información a través de consultas que se deben realizar con sentencias de SQL, en vez de esto, la búsqueda en una base de datos documental se la realiza entre los documentos que contenga la base, y por medio de algún campo clave se localiza el documento al que deseo visualizar con su respectiva información.

### **3.3. Diagrama de Clases**

#### **3.3.1 Conceptos básicos**

**Diagrama de clases del diseño.** Describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

**Métodos de acceso.** Son aquellos que recuperan o establecen el valor de los atributos...

#### **3.3.2. Diagrama de clases**

Este diagrama nos presenta las acciones o tareas que realizarán cada uno de los roles en el workflow.

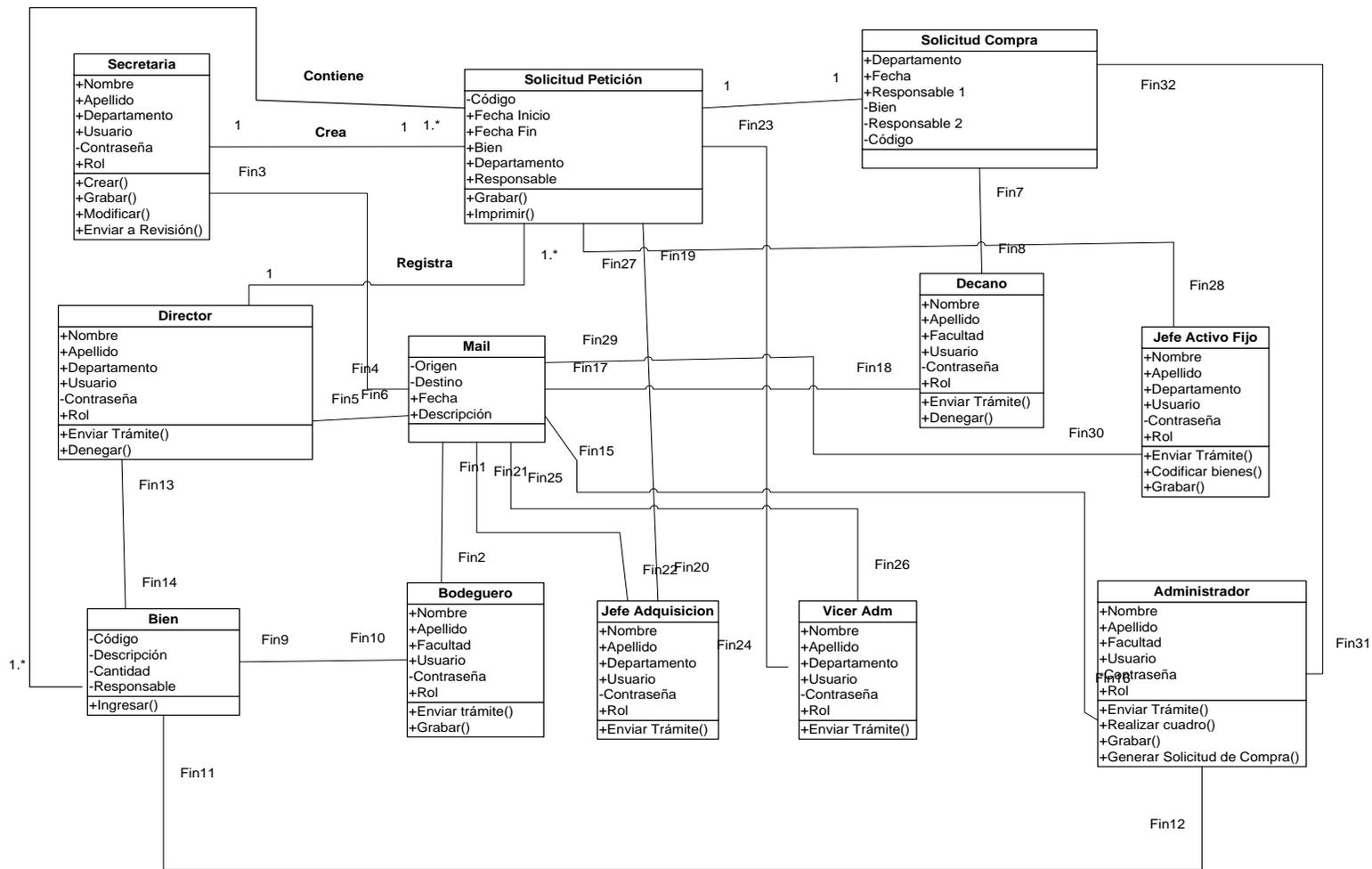


Figura 3.6 Diagrama de Clases

Fuente: Grupo de Proyecto

La secretaria **crea** la solicitud de petición y **registra** bienes, la solicitud de petición **contiene** bienes, el Director **registra** la solicitud de petición, el Decano **aprueba** solicitud de petición, el Administrador **revisa** la solicitud de petición y **genera** la solicitud de compra, el Vicerrector Administrativo y Jefe de Adquisición verifican la **solicitud de petición** y la **solicitud de compra**, el Jefe de Activo Fijo **codifica** los bienes y da **despacho** a la solicitud, el bodeguero registra los **bienes**, y en cada una de estas tareas que realizan los roles se enviará un mail automático para indicar en que estado se encuentra la solicitud de peticiones y cual es la siguiente acción que se debe tomar.

### **3.4 Desarrollo**

#### **3.4.1 Seguridades**

La seguridad está manejada en el workflow desde la Lista de Control de Acceso (ACL), en ella están creados los usuarios que acceden al Workflow, los roles que tendrán estos usuarios y qué tipo de usuario son, así como también dependiendo del rol que tenga cada uno de ellos, los permisos (tipo de acceso) que tendrán al manejar documentos.

Además de los permisos que se asignan dependiendo del rol, a cada usuario se puede también proporcionar permisos adicionales, sin

embargo para el desarrollo de este workflow no se está manejando esta opción porque la mayoría de los usuarios existentes necesitan las mismas características de acceso.

A continuación describimos cada uno de los roles creados y el tipo de acceso asignado a cada uno.

Tabla 22: Asignación de Roles y accesos

<b>Roles</b>	<b>Tipo de Acceso</b>
Secretaria	Editor
Secre Admin	Editor
Director	Editor
Decano	Editor
Administrador	Editor
Vicerrector	Editor
Jefe Adquis	Editor
Jefe Activo	Editor
Bodeguero	Editor

Fuente: Grupo de Proyecto

### **3.4.2 Descripción General del Desarrollo**

Para el desarrollo de este workflow hemos utilizado páginas, formularios, vistas, agentes, framesets, outlines, script libraries, Image Resource, Style Sheet , bases de datos:

### Bases de datos



Figura 3.7

Fuente: Aplicación Lotus Domino

En este workflow se utilizaron tres bases:

- SOLICITUD DE BIENES (SOLICITU.nsf), la cual contendrá las solicitudes creadas, aprobadas, y rechazadas.
- COMPARATIVO (COMPARAT.nsf), es utilizada para realizar el cuadro comparativo de cotizaciones e insertar los archivos adjuntos de las cotizaciones que anexasen.
- Inventario (Inventar.nsf), aquí se llevará el control del inventario de los bienes recibidos para el uso de la Facultad.

El detalle de los elementos está en el manual técnico del Workflow.

### **3.4.3 Códificación**

Para la programación se utilizó:

- Lenguaje de fórmula
- Lotus Script
- Java Script
- Código HTML

# **CAPITULO 4**

## **4. PRUEBAS**

### **4.1 Generalidades**

Ciertamente, la etapa de pruebas es muy importante en la elaboración de cualquier workflow, por lo que es aquí donde encontraremos fallas que podrían haber pasado inadvertidas durante su desarrollo. Es por eso, que se deben realizar varias pruebas antes de obtener la aplicación final, que sin lugar a dudas debe ser el resultado de un minucioso proceso de diferentes tipos de pruebas que perseguirán lograr la obtención de un workflow de calidad y que satisfaga las necesidades de los usuarios que harán uso de él.

### **4.2 Pruebas realizadas**

En la elaboración de este workflow, se llevaron a cabo diferentes tipos de pruebas, que a continuación detallamos:

- **Pruebas de Regresión:** con esta tipo de prueba se logró erradicar errores que se manifestaban por modificaciones en el diseño o en la codificación en ciertas partes del workflow, lo que provocaba que al complementar con nuevas partes elaboradas, algo de lo que hasta el momento estaba funcionando se pudiera ver afectado por los cambios recientes.  
Estas pruebas ayudaron en gran parte a corregir ese tipo de errores.
- **Pruebas de Estilo:** en este tipo de pruebas se busca verificar y corregir lo concerniente a tipos de letras, formato, tamaño, márgenes establecidos, etc.
- **Pruebas de Humo:** al realizar pruebas de humo pudimos corroborar si el workflow está apto para manejar gran cantidad de documentos de todos los usuarios involucrados, sin que se vuelva lento, ni sea difícil de utilizar por el tráfico causado.
- **Pruebas de GUI:** para realizar esta prueba, tuvimos que buscar la colaboración de personas ajenas al grupo de proyecto, mostrarles la aplicación y pedirles que le den uso, de esta forma pudimos obtener información acerca de si la interfaz es amigable, fácil de usar y que sin tener ningún conocimiento sobre el mismo, pueda ser utilizado sin problemas.

- **Pruebas de Desempeño:** este tipo de pruebas junto con las de regresión, son los dos tipos más importantes de pruebas que se realizaron para el workflow.

Los resultados de estas pruebas nos dieron información necesaria y vital como el poder saber hasta qué punto funcionaba el workflow, es decir si se logró o no conseguir los resultados deseados con la elaboración del mismo.

Todas estas pruebas dieron como resultado el poder obtener un workflow de calidad, fácil manejo y que se convertirá en una herramienta útil para cada uno de los usuarios que harán uso de él, aminorando además el tiempo empleado para cumplir con las funciones que debían realizar, facilitando así la labor de todos ellos y optimizando los recursos que se empleaban para realizar los procesos involucrados hasta ahora manualmente.

# **CAPITULO 5**

## **5. RECOMENDACIONES Y CONCLUSIÓN**

### **5.1 Recomendaciones**

#### **5.1.1 Hardware**

Para poder utilizar el workflow eficazmente se recomienda tener disponible los siguientes componentes:

- Computador Pentium/AMD de 1.1 Ghz. en adelante
- 256 de memoria RAM mínimo
- Unidad CD-RW (Opcional)
- Tarjeta de red 10/100
- Impresora (Opcional, puede ser una impresora en red para optimizar recursos)
- Scanner (Para la Administración de la Facultad)

### **5.1.2 Software**

El software recomendado es el siguiente:

- Sistema Operativo Windows XP o sus posteriores
- Microsoft Office, herramienta para manejo de hojas electrónicas
- Lotus Domino 6.5 mínimo (Para el Administrador de la herramienta)

Como recomendación general para el uso del workflow, podemos citar que debido a que su uso está orientado a aminorar el tiempo que se utiliza actualmente para la petición de bienes de una Facultad, lo óptimo sería que cada rol que lo utilice no demore mucho tiempo en enviar el trámite al siguiente rol del flujo.

## **5.2 Conclusión**

La realización de este workflow fue el resultado de analizar la problemática existente en una Facultad de la Universidad de Guayaquil al momento de necesitar bienes y hacer la petición a Administración Central para que realice la compra.

Conociendo esta situación, fue necesario establecer un mecanismo por el cual se corrijan todo tipo de falencias que actualmente ocurren en este proceso, y al mismo tiempo dar al usuario una herramienta fácil de manejar, que simplifique todas las acciones que deba realizar para este proceso.

El workflow está estructurado de tal forma que puede ser utilizado para cualquier flujo de procesos en el que se realicen peticiones y éstas deban ser aprobadas por una serie de instancias antes de suplir la necesidad del solicitante.

Es por esto que estamos seguros que nuestro “Workflow para Petición de Bienes en una Facultad”, es la solución eficaz para llevar a cabo todos y cada uno de los pasos a seguir para lograr que este tipo de flujos sean realizados y concluidos en un corto tiempo, logrando así la satisfacción de todos los usuarios que hagan peticiones a cualquier institución.

# INDICE GENERAL

<b>CAPITULO I</b>	<b>Pág</b>
<b>1.1. MANUAL DE USUARIO</b>	
1.1.1. ¿Cómo realizar una nueva Solicitud de Petición?	1
1.1.2. ¿Cómo Editar una solicitud ya creada?	5
1.1.3. ¿Cómo verificar solicitudes?	6
1.1.4. ¿Cómo denegar solicitudes?	7
1.1.5. ¿Cómo aprobar solicitudes?	8
1.1.6. ¿Cómo realizar el Cuadro Comparativo de Cotizaciones?	9
1.1.7. ¿Cómo realizar la Solicitud de Compra?	12
1.1.8. ¿Cómo realizar la codificación de los bienes?	14
1.1.9. ¿Cómo asignar los bienes a usuario solicitante?	15
1.1.10. ¿Cómo realizar la recepción de los bienes en el Workflow?	16
1.1.11. ¿Cómo imprimir un documento?	17

<b>CAPITULO II</b>	<b>Pág</b>
<b>2.1. Manual técnico de la herramienta Lotus Domino</b>	<b>18</b>
2.1.1. Servidor	18
2.1.1.1. Instalación del servidor	18
2.1.1.2. Configuración del servidor	25
2.1.2. Cliente	34
2.1.2.1. Instalación del cliente	34
2.1.2.2. Configuración del cliente	40
2.1.3. Configuraciones de Lotus Domino Administrador	45
2.1.4. Creación de las bases de datos y elementos de diseño	52
<b>2.2. Manual técnico del Workflow Petición de Bienes</b>	<b>54</b>
<b>2.2.1. Base SOLICITU.nsf</b>	<b>55</b>
2.2.1.1. Creación de frameset	55
2.2.1.2. Creación de páginas	57
2.2.1.3. Creación de forms	67
2.2.1.4. Creación de view	79
2.2.1.1. Creación de shared code	81
<b>2.2.2. Base COMPARAT.nsf</b>	<b>97</b>
2.2.2.1. Creación de forms	97
2.2.2.2. Creación de view	99

<b>2.2.3. Base Inventar.nsf</b>	101
2.2.3.1. Creación de forms	101
2.2.3.2. Creación de view	103
2.2.3.3. Creación de shared code	105
<b>2.3. Creación de shared resource para todas las bases</b>	119
2.3.1. Peticiones.css	120
2.3.2. View.css	127
<b>2.4. Accesos y Permisos de los Usuarios</b>	130
2.4.1. ACL	130
2.4.2. Roles	131

# INDICE GENERAL DE FIGURAS

<b>Capítulo 1</b>	<b>Pág</b>
Figura 1.1. Registrando Usuario y clave para ingresar	1
Figura 1.2. Bienvenida al workflow	2
Figura 1.3. Creación de una nueva solicitud	2
Figura 1.4. Ingreso de bienes	3
Figura 1.5. Formulario de Bienes	4
Figura 1.6. Botón Guardar	4
Figura 1.7. Botón Nuevo	4
Figura 1.8. Botón Regresar	5
Figura 1.9. Botón Enviar	5
Figura 1.10. Editando la solicitud	5
Figura 1.11. Botones Editar y Guardar	6
Figura 1.12. Solicitudes Pendientes en Facultad	7
Figura 1.13. Solicitudes para verificar	8
Figura 1.14. Botón Denegar	8
Figura 1.15. Verificación de Solicitudes para aprobar	9
Figura 1.16. Botón Enviar	9
Figura 1.17. Botón para realizar el Cuadro de Cotizaciones	9
Figura 1.18. Cuadro comparativo	10

	<b>Pág</b>
Figura 1.19. Botón Totales#	11
Figura 1.20. Botón Comparar	11
Figura 1.21. Anexos adjuntos	12
Figura 1.22. Botones para guardar y regresar	12
Figura 1.23. Botón Generar Solicitud	12
Figura 1.24. Formulario de Solicitud de Compra	13
Figura 1.25. Botones Grabar y Regresar	13
Figura 1.26. Botón Enviar	14
Figura 1.27. Codificación de bienes	14
Figura 1.28. Botón Guardar	15
Figura 1.29. Botón Denegar	15
Figura 1.30. Asignación de Bienes	16
Figura 1.31. Recepción de Bienes	16
Figura 1.32. Botón Finalizar	17
Figura 1.33. Botón Imprimir	17

<b>CAPITULO 2</b>	<b>Pág</b>
Figura 2.1. Carpeta de ubicación del ícono de instalación	18
Figura 2.2. Ícono de instalación del Servidor de Lotus Domino	19
Figura 2.3. Ventana de ubicación de los archivos extraídos	19
Figura 2.4. Ventana de extracción de archivos	20
Figura 2.5. Ventana del setup de instalación de Lotus Domino	20
Figura 2.6. Ventana de instalación de Lotus Domino	21
Figura 2.7. Ventana de aceptación de términos de Lotus Domino	21
Figura 2.8. Ventana de registro de Lotus Domino	22
Figura 2.9. Ventana de ubicación de Lotus Domino	22
Figura 2.10. Ventana de selección de tipo de servidores Lotus Domino	23
Figura 2.11. Ventana de ubicación de los archivos de Lotus Domino	23
Figura 2.12. Ventana de instalación y copias de archivos de Lotus Domino	24
Figura 2.13. Ventana de instalación final de Lotus Domino	24
Figura 2.14. Configuración del Servidor de Lotus Domino	25
Figura 2.15. Ventana de selección de servicios de carga de Lotus Domino	26
Figura 2.16. Ventana de carga de Lotus Domino	26
Figura 2.17. Ventana de Bienvenida de Configuración de Lotus Domino	27
Figura 2.18. Ventana de selección del tipo de Servidor a configurar	28
Figura 2.19. Especificación de nombre y título del Servidor Lotus Domino	28
Figura 2.20. Configuración del nombre y clave de la Organización	29
Figura 2.21. Especificación del Domino Domain Name	29
Figura 2.22. Registro de Nombre y clave del Administrador	30
Figura 2.23. Especificación de las tareas el Servidor	30

	<b>Pág</b>
Figura 2.24. Seleccionando la red de trabajo	31
Figura 2.25. Opciones avanzadas de red	31
Figura 2.26. Seguridades del Servidor	32
Figura 2.27. Resumen de configuración del Servidor	32
Figura 2.28. Server Setup	33
Figura 2.29. Fin de la configuración del servidor	33
Figura 2.30. Carpeta de ubicación del ícono de instalación del Cliente	34
Figura 2.31. Ícono de instalación del Cliente de Lotus Notes	34
Figura 2.32. Ubicación para grabar los archivos	35
Figura 2.33. Extracción de archivos a la carpeta destino	35
Figura 2.34. Bienvenida a la Instalación de Lotus Notes	36
Figura 2.35. Aceptación de términos de la herramienta	36
Figura 2.36. Registro de información del usuario y organización	37
Figura 2.37. Selección de ruta destino del Lotus Notes	37
Figura 2.38. Seleccionando los programas que desee instalar	38
Figura 2.39. Ventana de Instalación lista para empezar la tarea	38
Figura 2.40. Instalación de Lotus Notes y demás herramientas	39
Figura 2.41. Fin de la instalación de Lotus Notes	39
Figura 2.42. Levantando el Servidor Domino	40
Figura 2.43. Tipo de servicio a levantar	40
Figura 2.44. Tareas levantadas en el servidor	41
Figura 2.45. Abriendo Lotus Notes	41
Figura 2.46. Cargando Lotus Notes	42

	<b>Pág</b>
Figura 2.47. Bienvenida a la configuración de Lotus Notes	42
Figura 2.48. Registro de información del usuario y servidor	43
Figura 2.49. Ingreso de clave para registrar la conexión con el servidor	43
Figura 2.50. Configuración completa	44
Figura 2.51. Ventana de trabajo de Lotus Notes	44
Figura 2.52. Registro de usuarios	45
Figura 2.53. Ingreso de clave	45
Figura 2.54. Información de la certificación para la creación de usuarios	46
Figura 2.55. Registro de usuarios nuevos	47
Figura 2.56. Ubicación de los Ids	47
Figura 2.57. Creación de localidades	48
Figura 2.58. Creando un usuario nuevo	49
Figura 2.59. Configurando el servidor	50
Figura 2.60. Configurando las seguridades del servidor	51
Figura 2.61. Configurando el tab de Internet	51
Figura 2.62. Ejemplos de bases de datos	52
Figura 2.63. Creación de bases de datos	53
Figura 2.64. Base de dato creada	53
Figura 2.65. Bases de datos diseñadas para el workflow	55
Figura 2.66. Creación de frameset en una aplicación	57
Figura 2.67. Menú Create con la lista de opciones	60
Figura 2.68. Creación de páginas domino	66
Figura 2.69. Creación de formularios	68

	<b>Pág</b>
Figura 2.70. Vista de Solicitudes creadas	80
Figura 2.71. Creación de un outline	87
Figura 2.72. Formulario cuadro comparativo	99
Figura 2.73. Vista de Solicitudes	100
Figura 2.74. Creación del formulario inventario	101
Figura 2.75. Vista de Inventario	104
Figura 2.76. Permisos del ACL	131
Figura 2.77 Roles asignados a los usuarios	132

# INDICE DE TABLAS

	<b>Pág</b>
Formulario Bienes	
Tabla 2.1. Campos usados	70
Tabla 2.2. Acciones	71
Formulario Departamentos	
Tabla 2.3. Campos usados	71
Tabla 2.4. Acciones	71
Formulario Roles	
Tabla 2.5. Campos usados	72
Tabla 2.6. Acciones	72
Formulario Solicitud de Compra	
Tabla 2.7. Campos usados	73
Tabla 2.8. Acciones	73
Tabla 2.9. Secciones y Vistas Embebidas	74
Formulario Solicitudes	
Tabla 2.10. Campos usados	74
Tabla 2.11. Acciones	75
Tabla 2.12. Secciones y Vistas Embebidas	76
Formulario \$\$View Template for default	
Tabla 2.13. Vistas Embebidas	76
Formulario \$\$View Template for Departamentos	
Tabla 2.14. Vistas Embebidas	77

	<b>Pág</b>
Formulario \$\$View Template for MailUsuarios	
Tabla 2.15. Vistas Embebidas	77
Formulario \$\$View Template for Mis Solicitudes	
Tabla 2.16. Vistas Embebidas	77
Formulario \$\$View Template for Solicitud de Compra	
Tabla 2.17. Vistas Embebidas	77
Formulario \$\$View Template for Solicitudes en Admin. Central	
Tabla 2.18. Vistas Embebidas	78
Formulario \$\$View Template for Solicitudes en Proceso Facultad	
Tabla 2.19. Vistas Embebidas	78
Formulario \$\$View Template for Solicitudes	
Tabla 2.20. Vistas Embebidas	78
Formulario \$\$View Template for Trámites Finalizados	
Tabla 2.21. Vistas Embebidas	79
Formulario \$\$View Template for Trámites Rechazados	
Tabla 2.22. Vistas Embebidas	79
Tabla 2.23. Vista Creadas	80
Formulario Solicitudes_Comparativo	
Tabla 2.24. Campos usados	98
Tabla 2.25. Acciones	98
Formulario \$View Template for Mis Solicitudes	
Tabla 2.26. Vistas Embebidas	99
Tabla 2.27. Vista comparativo	100

	<b>Pág</b>
Formulario Inventario	
Tabla 2.28. Campos usados	102
Tabla 2.29. Acciones	103
Tabla 2.30 \$\$View Template for Inventario	103
Tabla 2.31. Vistas Inventario	104
Tabla 2.32. Usuarios Creados con Roles Asignados	132

# Capítulo 1

## 1.1. MANUAL DE USUARIO

### 1.1.1. ¿Cómo realizar una nueva Solicitud de Petición?

1. Ingrese al workflow (flujo de procesos) con su **Usuario** y **Contraseña**, en la ventana que se muestra a continuación, una vez ingresados los datos de clic en el botón **“Aceptar”**

Registrando Usuario y clave para ingresar

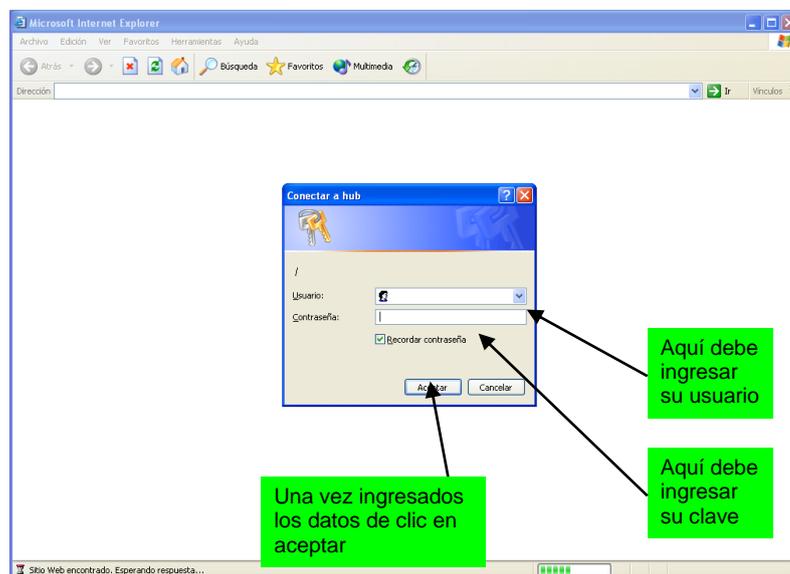


Figura 1.1 Fuente: Grupo de Proyecto

## 2. Cuando haya ingresado al workflow aparece la ventana de trabajo.

Bienvenida al workflow

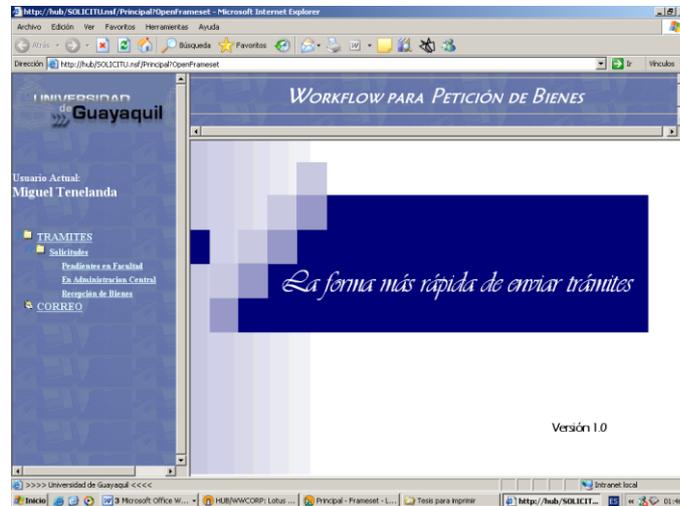


Figura 1.2. Fuente: Grupo de Proyecto

## 3. De clic en la opción “Nueva” para realizar la nueva solicitud.

Creación de una nueva solicitud

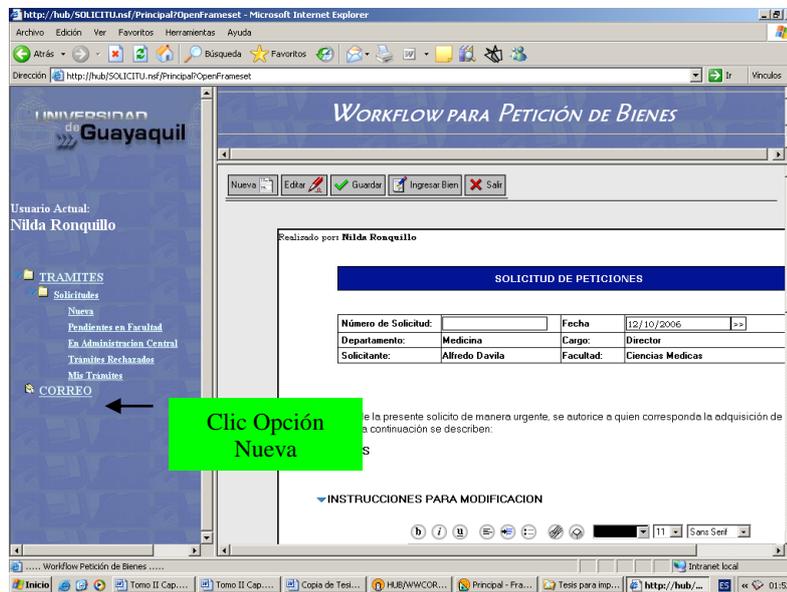


Figura 1.3 Fuente: Grupo de Proyecto

4. Ingrese los Bienes a solicitar haciendo clic en el botón **“Ingresar Bien”**.

Ingreso de bienes

WORKFLOW DE PETICION DE BIENES - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://hub/SOLICITU.nsf/Solicitudes?OpenForm>

Nueva Guardar Ingresar Bien Imprimir Salir

Realizado por: **Nilda Ronquillo**

**SOLICITUD DE PETICIONES**

Número de Solicitud:	<input type="text"/>	Fecha:	07/10/2006
Departamento:	Medicina	Cargo:	Director
Solicitante:	Alfredo Davila	Facultad:	Ciencias Medicas

Por medio de la presente solicito de manera urgente, se autorice a quien corresponda la adquisición de los siguientes bienes que a continuación se describen:

- ▶ BIENES
- ▶ COTIZACIONES

Listo Intranet local

Figura 1.4 Fuente: Grupo de Proyecto

5. Aparece el formulario de Ingreso de Bienes, donde debe seleccionar el tipo de bien, ingresar la descripción y cantidad de el/los bienes requeridos.

## Formulario de Bienes

Figura 1.5

Fuente: Grupo de Proyecto

- Ya ingresado los datos al sistema debe guardar los datos haciendo clic en el botón **“Guardar”**.

Botón Guardar



Figura 1.6 Fuente: Grupo de Proyecto

- Si se desea ingresar un nuevo Bien solo tiene que hacer clic en el botón **“Nuevo”** en el formulario de Bienes a Solicitar y repetir el paso anterior.

Botón Nuevo



Figura 1.7 Fuente: Grupo de Proyecto

8. Cuando se hayan ingresado todos los bienes a solicitar debe hacer clic en el botón **“Regresar”** para regresar a la Solicitud de Petición.

Botón Regresar



Figura 1.8 Fuente: Grupo de Proyecto

9. Finalmente grabe la solicitud y haga clic en el botón **“Enviar”** para que la solicitud sea enviada al siguiente usuario responsable.

Botón Enviar



Figura 1.9 Fuente: Grupo de Proyecto

### 1.1.2. ¿Cómo Editar una solicitud ya creada?

1. Ingrese al workflow, como se indico anteriormente, y seleccione la opción **“Mis Trámites”** y seguido la solicitud que desea modificar.

Editando la solicitud

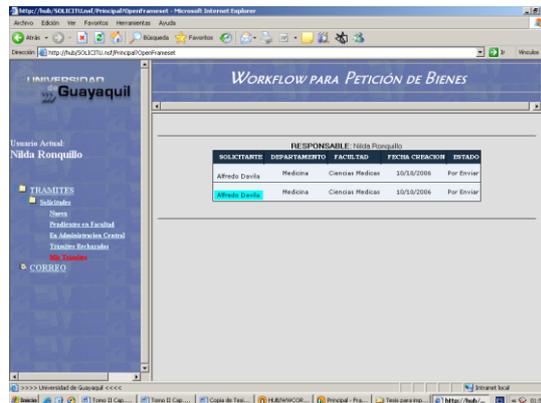


Figura 1.10 Fuente: Grupo de Proyecto

- Haga clic en el botón **“Editar”**, y modifique los campos que desee editar y guarde los cambios haciendo clic en el botón **“Guardar”**

Botones Editar y Guardar

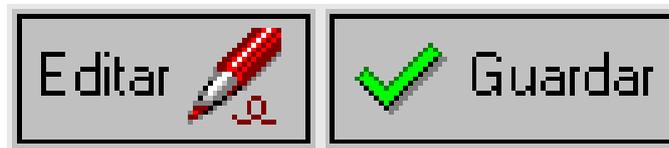


Figura 1.11 Fuente: Grupo de Proyecto

### 1.1.3. ¿Cómo verificar solicitudes?

- Accese al workflow con su usuario y contraseña.

2. Si quiere verificar las solicitudes que usted tiene pendiente, escoja la opción **“Mis Trámites”** y seleccione la solicitud que desea verificar.
3. Si quiere verificar si las solicitudes ya creadas aun se encuentran en la facultad, escoja la opción **“Pendientes en Facultad”**.
4. Si quiere verificar si las solicitudes ya creadas aun se encuentran en la facultad, escoja la opción **“En Administración Central”**.
5. Si quiere verificar si las solicitudes ya creadas han sido rechazadas, escoja la opción **“Trámites Rechazados”**.
6. Si quiere verificar si las solicitudes ya creadas han sido rechazadas, escoja la opción **“Trámites Finalizados”**.

#### Solicitudes Pendientes en Facultad

ESTADO	SOLICITANTE	DEPARTAMENTO	FACULTAD	FECHA CREACION	FECHA RECEPCION
Por Enviar	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	06/10/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	08/18/2006	
	Alfredo Davila	Medicina	Ciencias Medicas	08/18/2006	

Figura 1.12 Fuente: Grupo de Proyecto

### 1.1.4. ¿Cómo denegar solicitudes?

1. Accese al workflow con su usuario y contraseña.
2. Verifique las solicitudes pendientes en la opción **“Mis trámites”**, seleccione la que usted desee aprobar o denegar.
3. Una vez que haya accedido a la solicitud usted puede aprobarla o denegarla, haciendo clic en el botón correspondiente a la opción deseada.

#### Solicitudes para verificar

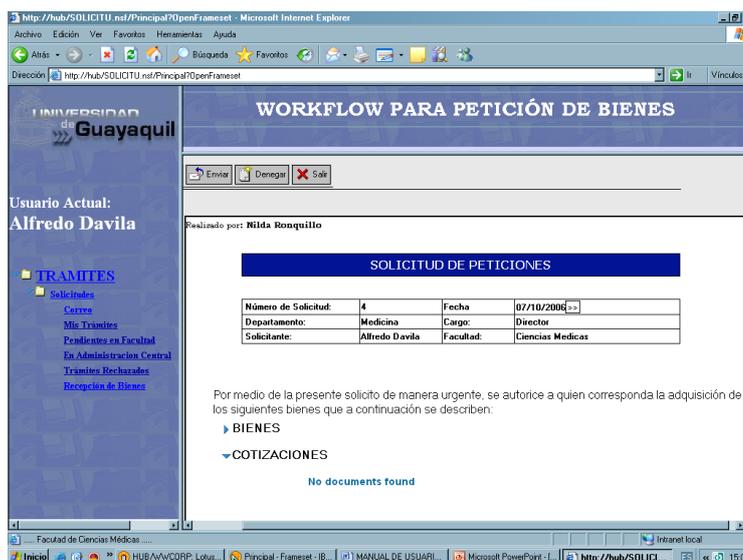


Figura 1.13 Fuente: Grupo de Proyecto

4. Si la opción es aprobarla haga clic en el botón **“Enviar”**, o si la opción es aprobarla haga clic en el botón **“Denegar”**

Botón Denegar



Figura 1.14 Fuente: Grupo de Proyecto

### 1.1.5. ¿Cómo aprobar solicitudes?

1. Accese al workflow con su usuario y contraseña.
2. Verifique las solicitudes pendientes en la opción **“Mis trámites”**, seleccione la que usted desee aprobar o denegar.
3. Una vez que haya accedido a la solicitud usted puede aprobarla o denegarla, haciendo clic en el botón correspondiente a la opción deseada.

Verificación de Solicitudes para aprobar

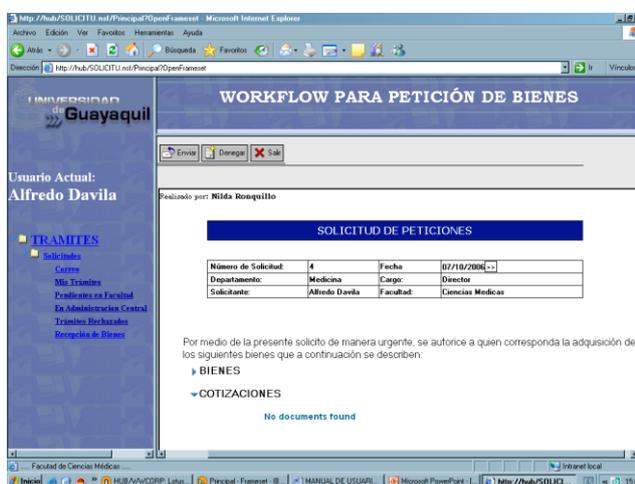


Figura 1.15 Fuente: Grupo de Proyecto

4. Si la opción es aprobarla haga clic en el botón **“Enviar”**, o si la opción es aprobarla haga clic en el botón **“Denegar”**

Botón Enviar



Figura 1.16 Fuente: Grupo de Proyecto

### 1.1.6. ¿Cómo realizar el Cuadro Comparativo de Cotizaciones?

1. Accese al workflow con su usuario y contraseña.
2. Verifique las solicitudes pendientes en la opción **“Mis trámites”**, seleccione la solicitud deseada.
3. Haga clic en el botón **“Cotizaciones”** para acceder al Formulario donde podrá realizar el Cuadro Comparativo de Cotizaciones.

Botón para realizar el Cuadro de Cotizaciones



Figura 1.17 fuente: Grupo de Proyecto

4. En el cuadro comparativo de cotizaciones ingrese los datos requeridos, tales como los nombres de las empresas, los bienes y los costos en que están valorados cada uno de los bienes por unidad.

Cuadro comparativo

Realizado por Jose Casales

**CUADRO COMPARATIVO DE COTIZACIONES**

Número de Solicitud: 123

Area Solicitante: Medicina

Empresa 1: \_\_\_\_\_

Empresa 2: \_\_\_\_\_

Empresa 3: \_\_\_\_\_

No-CANT.	DESCRIPCION	EMPRESA 1		EMPRESA 2		EMPRESA 3		Totales#
		P. UNIT.	P. TOTAL	P. UNIT.	P. TOTAL	P. UNIT.	P. TOTAL	
1								
	Sub-Totales							
	I.V.A. 12%							
	Total							

Figura 1.18 Fuente: Grupo de Proyecto

- Una vez ingresado un bien con su cantidad y valor correspondiente haga clic en el botón totales# que se encuentra a la derecha de cada fila de ingreso.

Botón Totales#



Figura 1.19 Fuente: Grupo de Proyectos

- Una vez que ingresó todos los datos, haga clic en el botón **“Comparar”** para obtener todos los resultados tales como subtotales, IVA, y totales por cada una de las empresas.

Botón Comparar



No CANT.	DESCRIPCION	EMPRESA 1		EMPRESA 2		EMPRESA 3	
		P. UNIT.	P. TOTAL	P. UNIT.	P. TOTAL	P. UNIT.	P. TOTAL
1			\$ 0,00		\$ 0,00		\$ 0,00
	Sub-Totales		\$ 0,00		\$ 0,00		\$ 0,00
	I.V.A. 12%		\$ 0,00		\$ 0,00		\$ 0,00
	Total		\$ 0,00		\$ 0,00		\$ 0,00

Cotización Ganadora	El mejor precio lo tiene cotización de
---------------------	--

Figura 1.20 Fuente: Grupo de Proyecto

7. Para anexar las cotizaciones previamente scaneadas y grabadas, haga clic en el botón **“Examinar”** para buscar la cotización selecciónela y luego haga clic en el botón **“Anexar”**.

Anexos adjuntos

Archivo Adjunto :	<input type="text"/>	Examinar...	Anexar
-------------------	----------------------	-------------	--------

Figura 1.21 Fuente: grupo de proyecto

8. Finalmente haga clic en el botón **“Guardar”** y en el botón **“Regresar”** para guardar los datos digitados y regresar a la solicitud de petición.

Botones para guardar y regresar



Figura 1.22 Fuente: Grupo de Proyecto

### 1.1.7. ¿Cómo realizar la Solicitud de Compra?

1. Ya realizado el Cuadro Comparativo de Cotizaciones, en la Solicitud de Petición haga clic en el botón “**Generar Solicitud**”

Botón Generar Solicitud



Figura 1.23 Fuente: Grupo de Proyecto

2. En el formulario de Solicitud de Compra ingrese los datos que justificarán, los bienes que se van a adquirir.

Formulario de Solicitud de Compra

Grabar Regresar Imprimir

Realizado por: **Jose Canelos**

### SOLICITUD DE COMPRA

Numero de Solicitud:	19	Fecha:	10/10/2006
Solicitante:	Alfredo Davila	Cargo:	Director
Departamento:	Medicina	Facultad:	Ciencias Medicas

▼ BIENES

Fecha	Tipo Bien	Descripción	Cantidad
09/10/2006	Equipos Eléctronicos sss		1

▼ SUMILLA

JUSTIFICATIVO	DESTINO DEL PEDIDO	TIEMPO DE VIDA BIEN
OBSERVACIONES	PREPARADO POR:	SOLICITADO POR:
	Dr. Wilson Maitta Mendoza Decano	Sr. José Canelos Escobar Administrador Encargado

Figura 1.24 Fuente: Grupo de Proyecto

3. Haga clic en el botón **“Grabar”** para guardar el documento, seguido del botón **“Regresar”** para regresar a la solicitud de petición.

Botones Grabar y Regresar



Figura 1.25 Fuente: Grupo de Proyectos

4. Una vez realizada el cuadro comparativo de cotizaciones, y la solicitud de compra, haga clic en el botón enviar, para que la documentación sea enviada al siguiente usuario responsable.

Botón Enviar



Figura 1.26 Fuente: grupo de Proyecto

### 1.1.8. ¿Cómo realizar la codificación de los bienes?

1. Ingrese al Workflow con su usuario y contraseña
2. En el formulario ingrese el código, seleccione el tipo de bien y la descripción de cada uno de los bienes solicitados.

Codificación de bienes

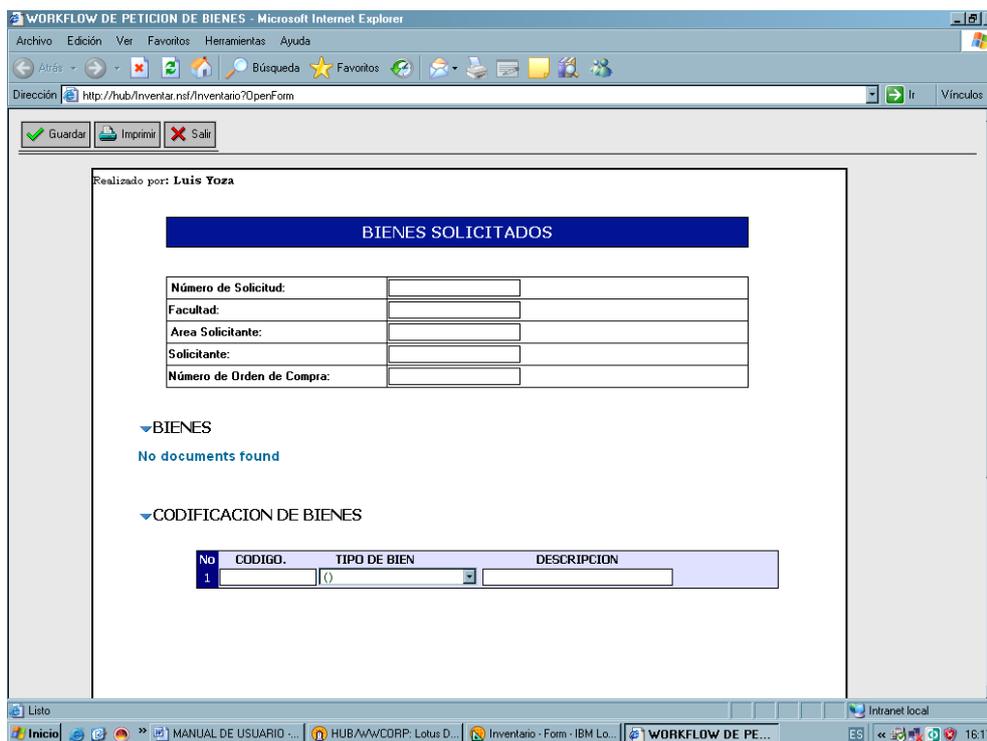


Figura 1.27 Fuente: grupo de Proyecto

- Luego que ingrese todos los bienes haga clic en el botón **“Guardar”**, para grabar el documento.

Botón Guardar



Figura 1.28 Fuente: Grupo de Proyecto

- Finalmente haga clic en el botón **“Despachar”** para que el documento sea enviado al siguiente usuario responsable.

Botón Denegar



Figura 1.29 Fuente: grupo de proyecto

### 1.1.9. ¿Cómo asignar los bienes a usuario solicitante?

1. Ingrese al Workflow con su usuario y contraseña
2. Haga clic en el botón **“Asignar”** e ingrese los datos correspondientes al formulario.

Asignación de Bienes

Realizado por: Luis Yoza

**BIENES SOLICITADOS**

Número de Solicitud:	143221
Facultad:	Ciencias Medicas
Area Solicitante:	Medicina
Solicitante:	Alfredo Davila
Número de Orden de Compra:	

▼ BIENES

Fecha	Tipo Bien	Descripción	Cantidad
10/10/2006	Equipos Electronicos	drdf	2

Figura 1.30 Fuente: Grupo de Proyecto

### 1.1.10. ¿Cómo realizar la recepción de los bienes en el Workflow?

1. Ingrese al Workflow con su usuario y contraseña
2. Seleccione la opción **“Recepción de bienes”**

#### Recepción de Bienes

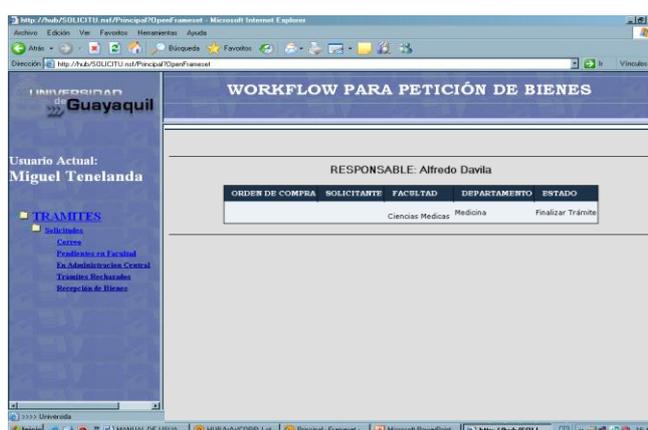


Figura 1.31 Fuente: Grupo de Proyecto

3. En el formulario haga clic en el botón **“Finalizar”** para registrar la recepción del bien.

#### Botón Finalizar

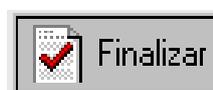


Figura 1.32 Fuente: Grupo de Proyecto

### 1.1.11. ¿Cómo imprimir un documento?

1. Para imprimir el documento solo debe hacer clic en el botón **“Imprimir”** y seleccionar la impresora con la que está trabajando.

Botón Imprimir



Figura 1.33 Fuente: Grupo de proyecto

# Capítulo 2

## 2.1. Manual técnico de la herramienta Lotus Domino

### 2.1.1. Servidor

#### 2.1.1.1. Instalación del servidor

Para la instalación del Servidor debe seguir los siguientes pasos

1. Haga doble clic en la carpeta Domino Server 6.5.2

Carpeta de ubicación del ícono de instalación



Domino Server  
6.5.2

Figura 2.1 Fuente: Instalación de Lotus Domino

2. Haga doble clic en el ícono **c56srna** para empezar a extraer los componentes para la instalación

Ícono de instalación del Servidor de Lotus Domino



Figura 2.2 Fuente: Instalación de Lotus Domino

3. Seleccione la carpeta donde quiere grabar los archivos, y presione el botón **Next**.

Ventana de ubicación de los archivos extraídos

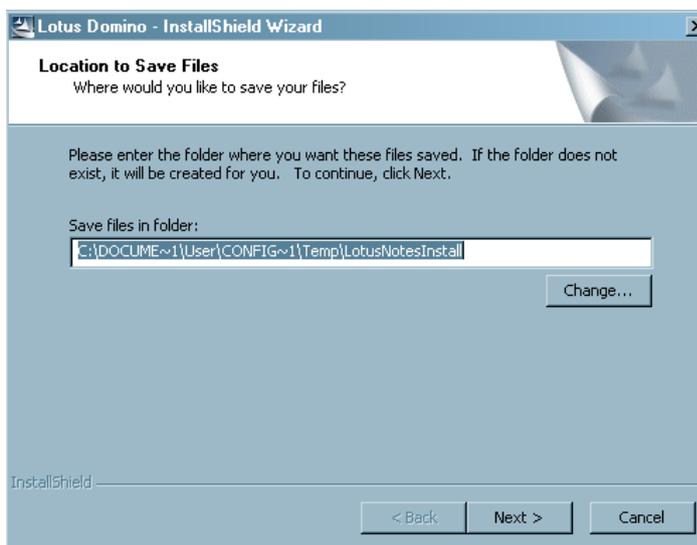


Figura 2.3 Fuente: Instalación de Lotus Domino

4. Empezara a extraer el contenido de los archivos al destino especificado

Ventana de extracción de archivos

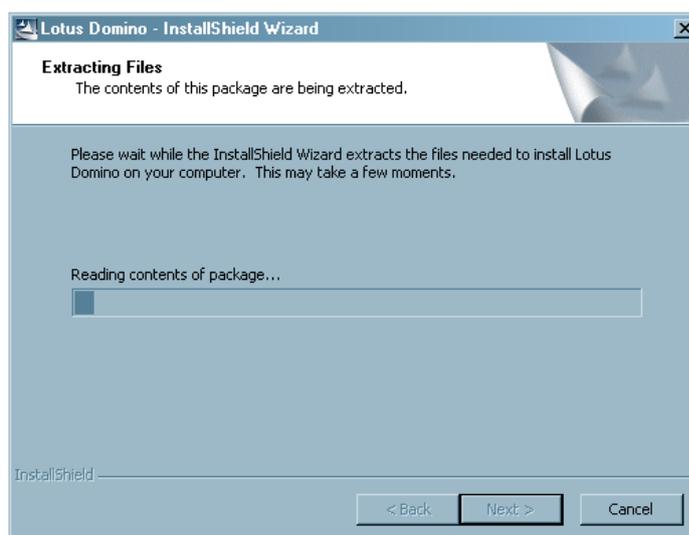


Figura 2.4 Fuente: Instalación de Lotus Domino

5. Saldrá el siguiente cuadro de texto el cual indica que se está preparando para la instalación.

Ventana del setup de instalación de Lotus Domino

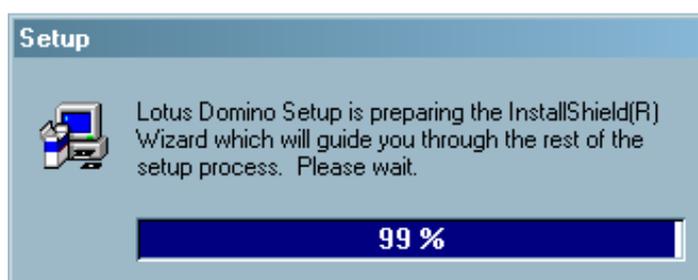


Figura 2.5 Fuente: Instalación de Lotus Domino

## 6. Haga clic en el botón “Next” para continuar la instalación

Ventana de instalación de Lotus Domino

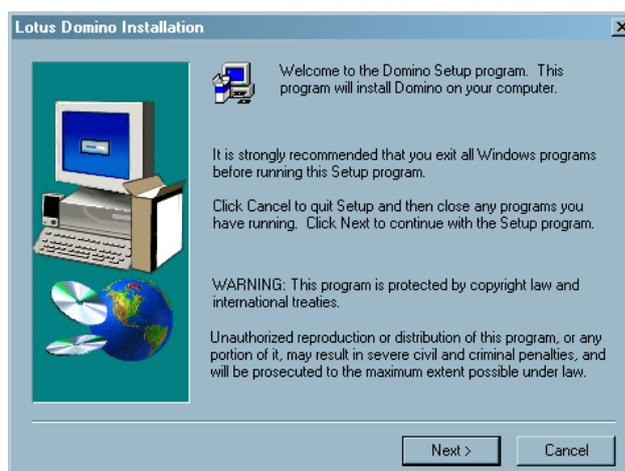


Figura 2.6 Fuente: Instalación de Lotus Domino

## 7. Aceptar los términos de la licencia del software

Ventana de aceptación de términos de Lotus Domino

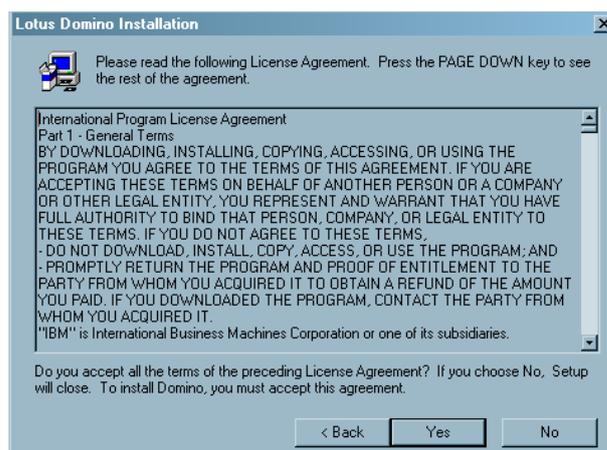


Figura 2.7 Fuente: Instalación de Lotus Domino

- Ingrese su nombre y el nombre de la compañía, y haga clic en el botón **“Next”**

Ventana de registro de Lotus Domino



Figura 2.8

Fuente: Instalación de Lotus Domino

- Seleccione el directorio donde se quiere instalar Lotus Domino, y haga clic en el botón **“Next”**

Ventana de ubicación de Lotus Domino

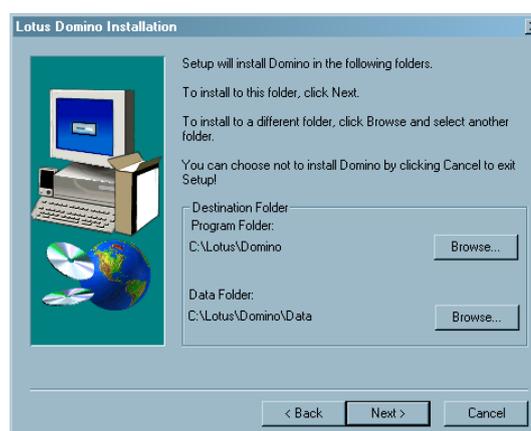


Figura 2.9

Fuente: Instalación de Lotus Domino

10. Seleccione el tipo de servidor que desee instalar, es recomendable que sea **“Domino Enterprise Server”** luego haga clic en el botón **“Next”**

Ventana de selección de tipo de servidores Lotus Domino

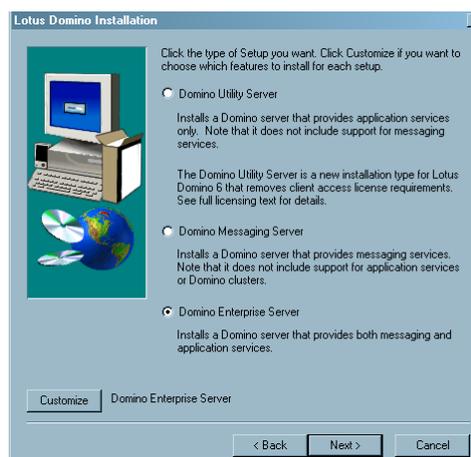


Figura 2.10 Fuente: Instalación de Lotus Domino

11. Seleccione el lugar y el nombre que tendrá la herramienta al momento de acceder a ella.

Ventana de ubicación de los archivos de Lotus Domino

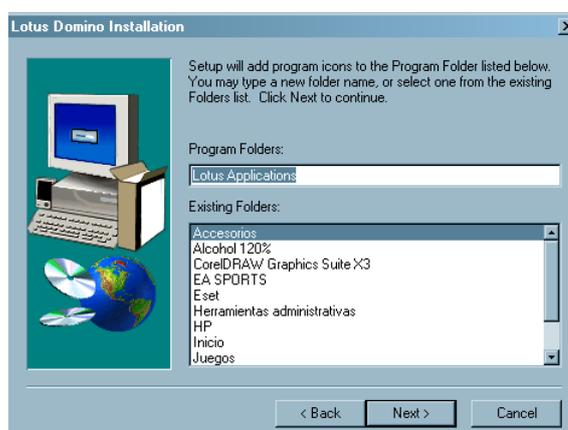


Figura 2.11 Fuente: Instalación de Lotus Domino

## 12. Comenzara a instalar la aplicación

Ventana de instalación y copias de archivos de Lotus Domino

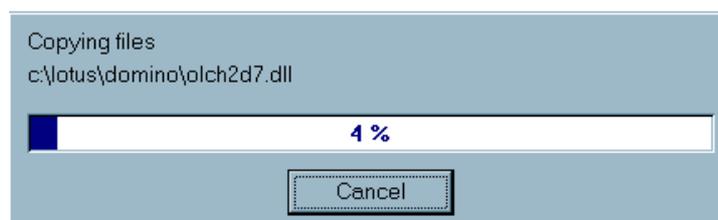


Figura 2.12

Fuente: Instalación de Lotus Domino

## 13. Haga clic en el botón **Finish** para finalizar la instalación

Ventana de instalación final de Lotus Domino



Figura 2.13

Fuente: Instalación de Lotus Domino

### 2.1.1.2. Configuración del servidor

1. Ingrese al menú Inicio, opción Programas y busque la opción Lotus Application, haga clic en la opción **Lotus Domino Server**

Configuración del Servidor de Lotus Domino



Figura 2.14

Fuente: Configuración de Lotus Domino

2. Seleccione el tipo de servicio con el cual se va a trabajar, se recomienda **Start Domino as a regular application** y haga clic en el botón **OK**

Ventana de selección de servicios de carga de Lotus Domino

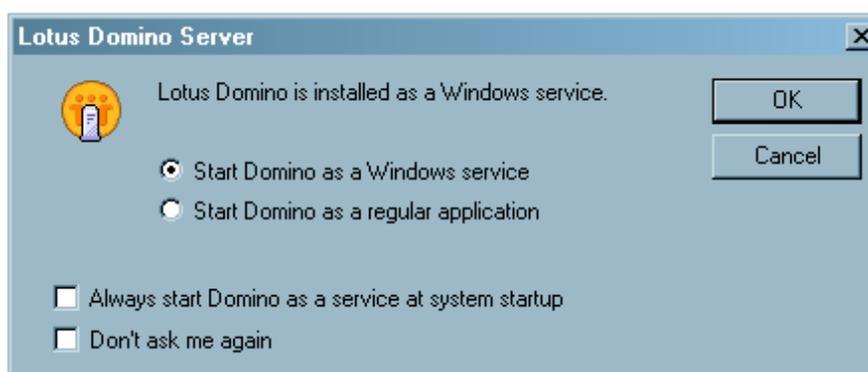


Figura 2.15

Fuente: Configuración de Lotus Domino

### 3. Empezara a cargar el servidor

Ventana de carga de Lotus Domino

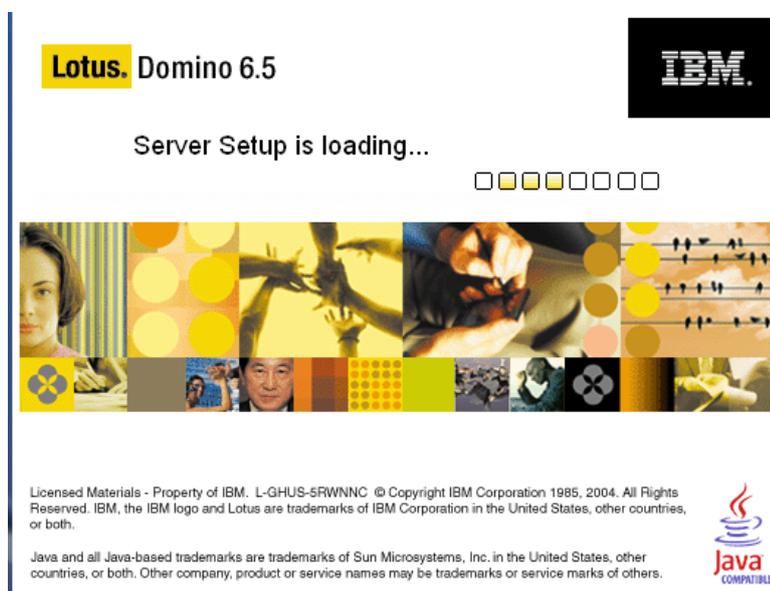


Figura 2.16

Fuente: Configuración de Lotus Domino

4. Saldrá el cuadro de diálogo de Bienvenida de Lotus, haga clic en el botón **Next**

Ventana de Bienvenida de Configuración de Lotus Domino

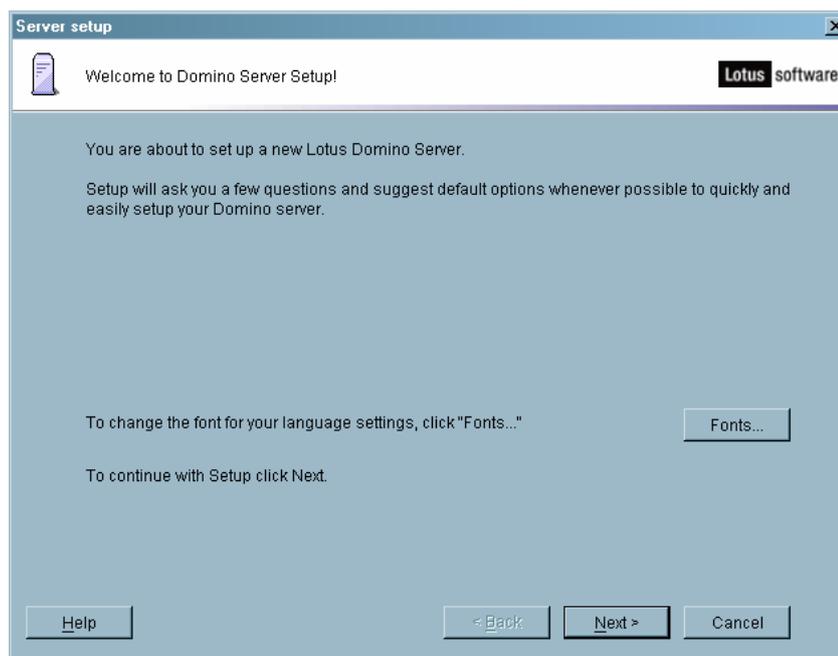


Figura 2.17

Fuente: Configuración de Lotus Domino

5. Seleccione si es el servidor principal a instalar o un servidor adicional. Si es el servidor principal que se está instalando seleccione la primera opción, haga clic en el botón **Next**.

## Ventana de selección del tipo de Servidor a configurar

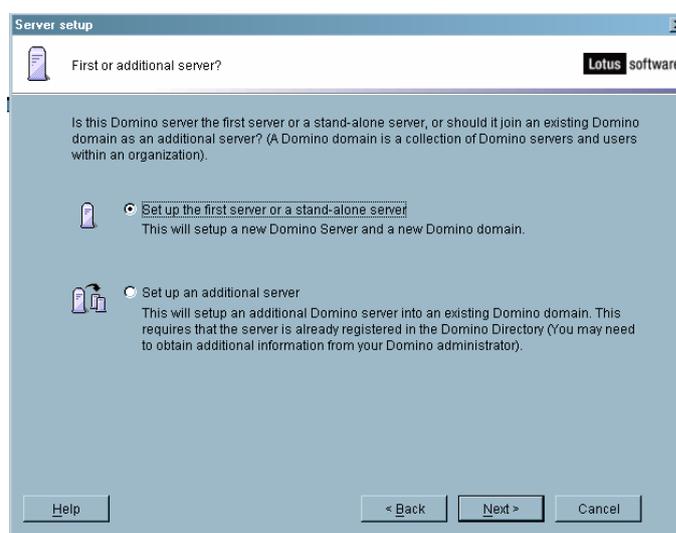


Figura 2.18 Fuente: Configuración de Lotus Domino

6. Ponga el nombre y título del servidor, haga clic en el botón **Next**

## Especificación de nombre y título del Servidor Lotus Domino

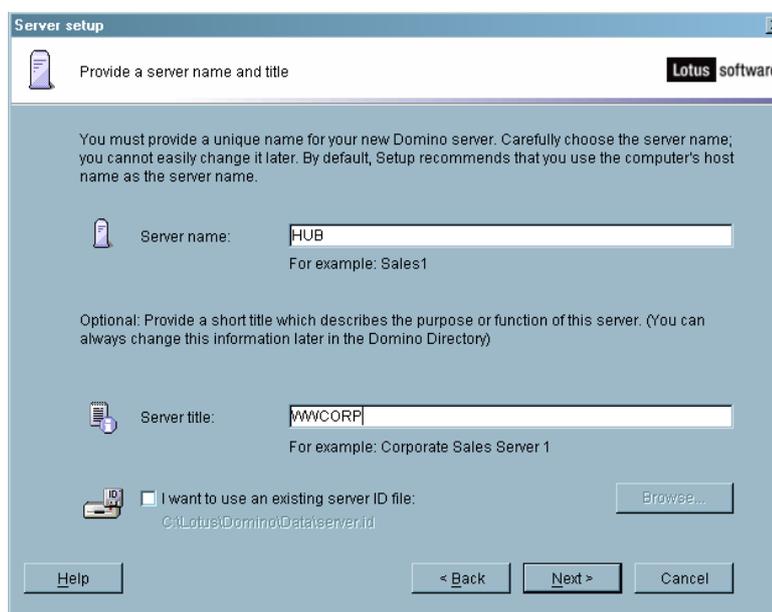


Figura 2.19 Fuente: Configuración de Lotus Domino

7. Escriba el nombre de la organización y su contraseña, haga clic en el botón **Next**

#### Configuración del nombre y clave de la Organización

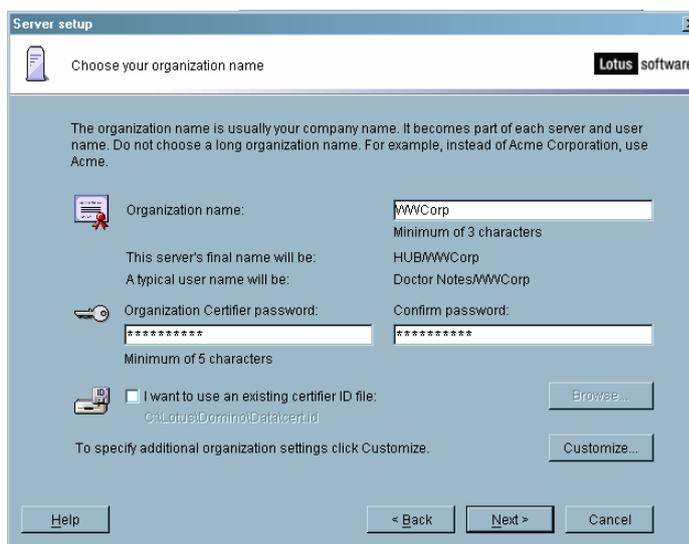


Figura 2.20 Fuente: Configuración de Lotus Domino

8. Escriba un dominio para el servidor, por lo general se recomienda que sea el mismo de la organización, haga clic en el botón **Next**

#### Especificación del Domino Domain Name

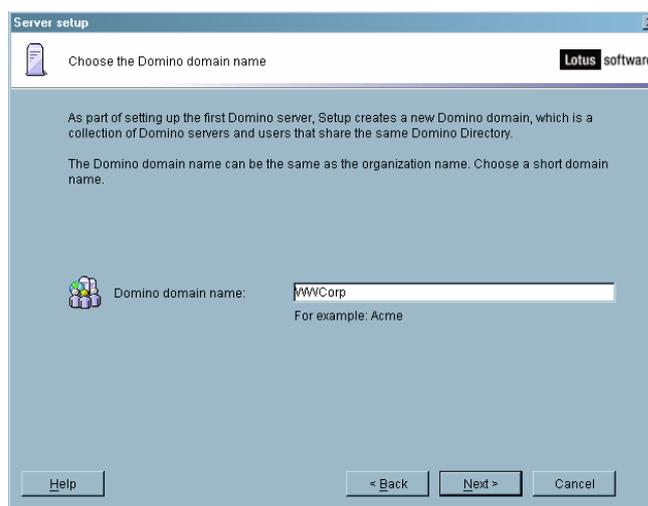


Figura 2.21 Fuente: Configuración de Lotus Domino

9. Especifique el nombre y contraseña del administrador, y seleccione la opción **Also save a local copy of the ID file**, haga clic en el botón **Next**

#### Registro de Nombre y clave del Administrador

The screenshot shows a 'Server setup' window with the title 'Specify an Administrator name and password'. It contains the following fields and options:

- First name:** Doctor
- Middle:** (empty)
- Last name (or generic account name):** Notes
- Administrator password:** (masked with asterisks)
- Confirm password:** (masked with asterisks)
- Minimum of 5 characters** (note below the password fields)
- The Administrator ID file will be stored inside the server's Domino Directory.**
- Also save a local copy of the ID file:** (with a 'Browse...' button and path C:\Lotus\Domino\Data\adminid)
- I want to use an existing Administrator ID file:** (with a 'Browse...' button and path C:\Lotus\Domino\Data\adminid)
- Buttons: Help, < Back, Next >, Cancel

Figura 2.22 Fuente: Configuración de Lotus Domino

10. Seleccione los servicios que proveerá su servidor y haga clic en el botón **Next**

#### Especificación de las tareas el Servidor

The screenshot shows a 'Server setup' window with the title 'What Internet services should this Domino Server provide?'. It contains the following options:

- Select the Internet services this Domino server will provide. Basic Notes and Domino services are set up by default. (You can always change these options later in the Domino Directory).**
- Setup Internet services for:**
  - Web Browsers (HTTP services)
  - Internet Mail Clients (SMTP, POP3 and IMAP services)
  - Directory services (LDAP services)
- To customize all other Domino services, click Customize. (with a 'Customize...' button)
- Buttons: Help, < Back, Next >, Cancel

Figura 2.23 Fuente: Configuración de Lotus Domino

11. Setee la red con la que trabajará su servidor, haga clic en el botón **Customize**

Seleccionando la red de trabajo

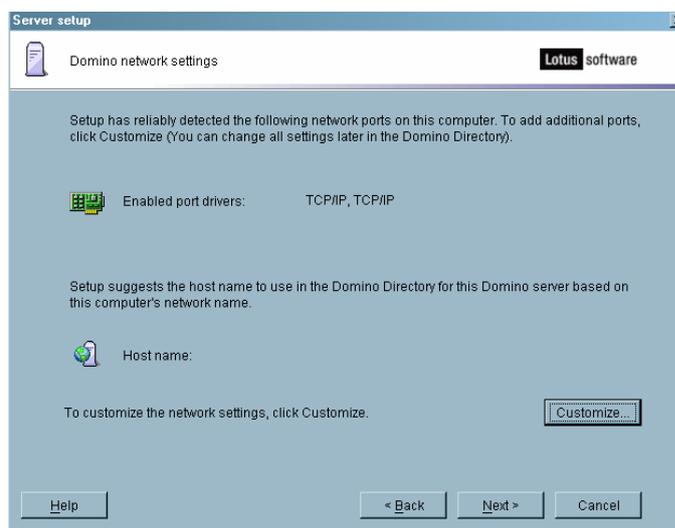


Figura 2.24 Fuente: Configuración de Lotus Domino

12. Ingrese el nombre del servidor seguido del Domino Domain Name, haga clic botón **OK** y luego en el botón **Next**

Opciones avanzadas de red

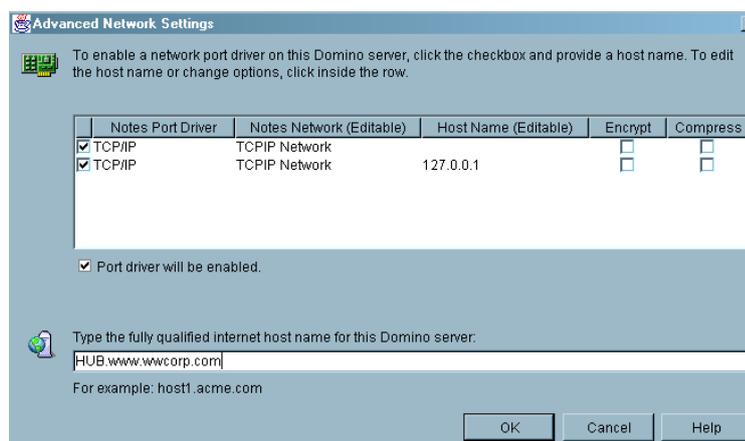


Figura 2.25 Fuente: Configuración de Lotus Domino

13. Seleccione las casillas para activar las medidas de seguridad de acceso al servidor.

#### Seguridades del Servidor

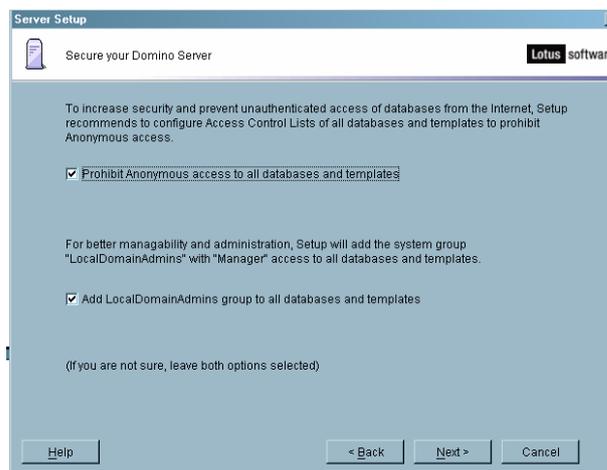


Figura 2.26 Fuente: Configuración de Lotus Domino

14. Saldrá una pantalla con el resumen de configuración del Servidor.

#### Resumen de configuración del Servidor

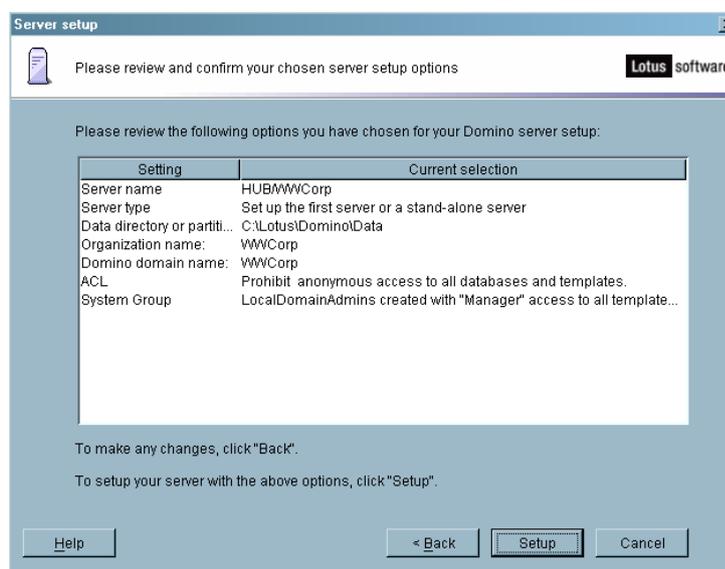


Figura 2.27 Fuente: Configuración de Lotus Domino

## 15. Haga clic en el botón **Setup**

Server Setup

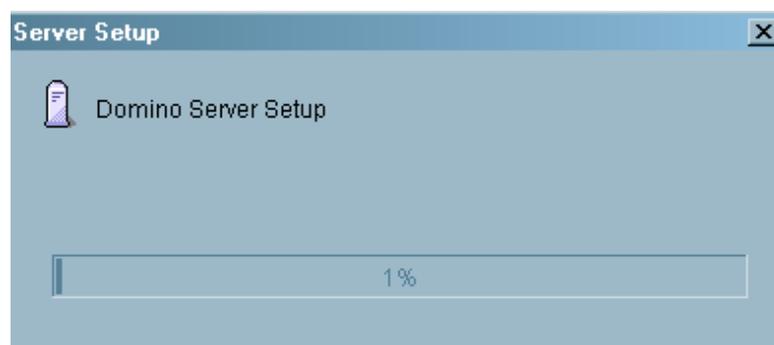


Figura 2.28 Fuente: Configuración de Lotus Domino

## 16. Finalmente haga clic en el botón **Finish** para terminar la configuración.

Fin de la configuración del servidor

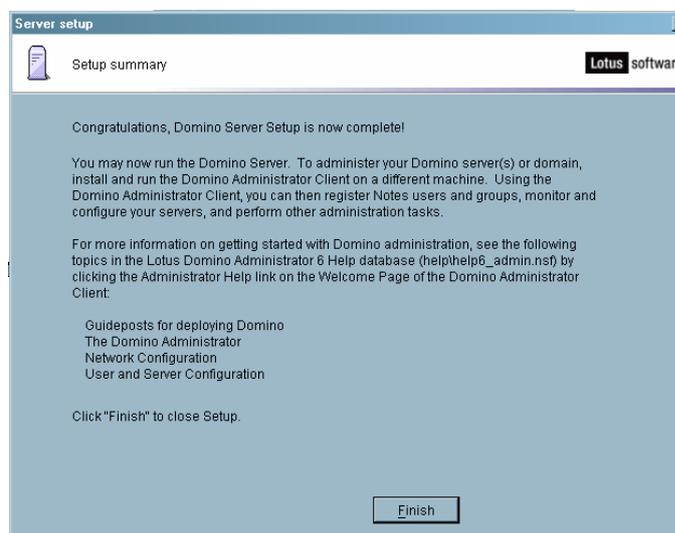


Figura 2.29 Fuente: Configuración de Lotus Domino

## 2.1.2. Cliente

### 2.1.2.1. Instalación del cliente

Para la instalación del Cliente debe seguir los siguientes pasos:

1. Haga doble clic en la carpeta Lotus Cliente 6.5.2

Carpeta de ubicación del ícono de instalación del Cliente



Figura 2.30 Fuente: Instalación de Lotus Notes

2. Haga doble clic en el ícono **c56sqna** para empezar a extraer los componentes para la instalación

Ícono de instalación del Cliente de Lotus Notes



Figura 2.31 Fuente: Instalación de Lotus Notes

3. Seleccione la carpeta donde quiere grabar los archivos, y presione el botón **Next**.

## Ubicación para grabar los archivos

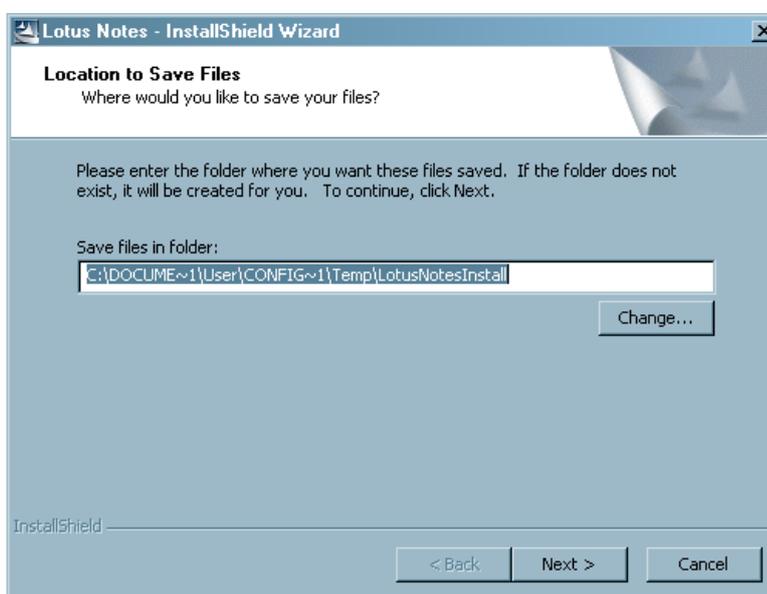


Figura 2.32 Fuente: Instalación de Lotus Notes

4. Empezara a extraer el contenido de los archivos al destino especificado

## Extracción de archivos a la carpeta destino

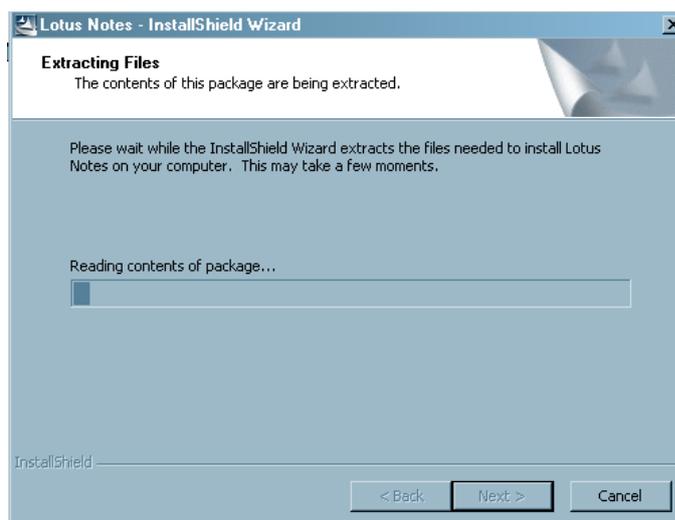


Figura 2.33 Fuente: Instalación Lotus Notes

5. Saldrá la ventana de bienvenida de Lotus Notes, haga clic en el botón **“Next”**

Bienvenida a la Instalación de Lotus Notes

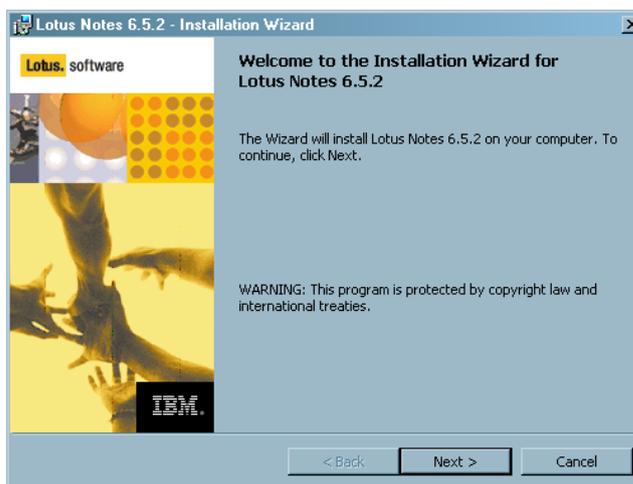


Figura 2.34 Fuente: Instalación Lotus Domino

6. Acepte lo términos de la licencia para continuar la instalación de Lotus Notes, haga clic en el botón **“Next”**

Aceptación de términos de la herramienta



Figura 2.35 Fuente: Instalación de Lotus Domino

7. Ingrese la información sobre el usuario y la compañía, haga clic en el botón **“Next”**

Registro de información del usuario y organización

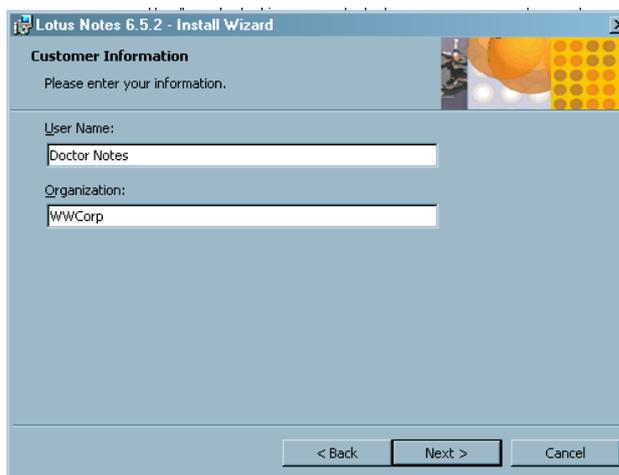


Figura 2.36 Fuente: Instalación de Lotus Notes

8. Seleccione la ruta donde desee instalar la herramienta, haga clic en el botón **“Next”**

Selección de ruta destino del Lotus Notes

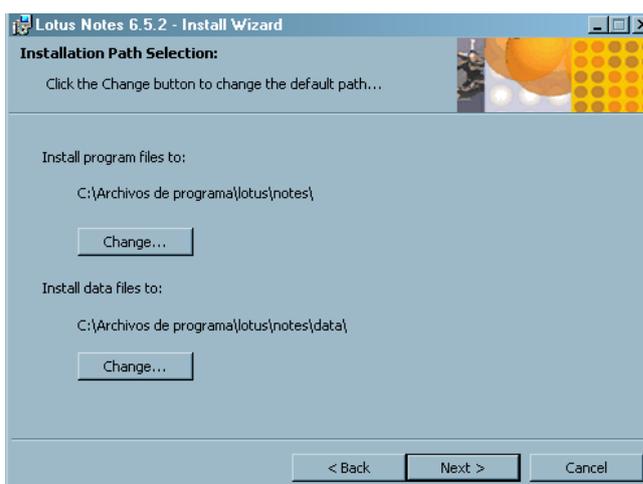


Figura 2.37 Fuente: Instalación de Lotus Notes

9. Seleccione los servicios con los cuales desee trabajar, y haga clic en el botón **“Next”**

Seleccionando los programas que desee instalar

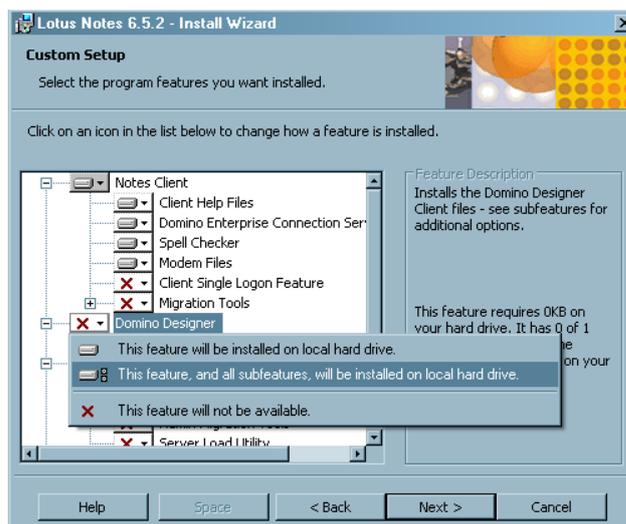


Figura 2.38 Fuente: Instalación Lotus Notes

10. Aparece la ventana que indica que la configuración está lista y haga clic en el botón **“Install”**

Ventana de Instalación lista para empezar la tarea

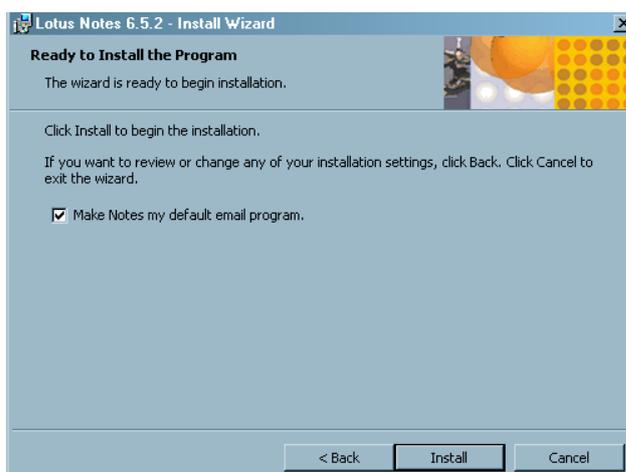


Figura 2.39 Fuente: Instalación Lotus Notes

## 11. Comenzará a realizarse la instalación de Lotus Notes

Instalación de Lotus Notes y demás herramientas

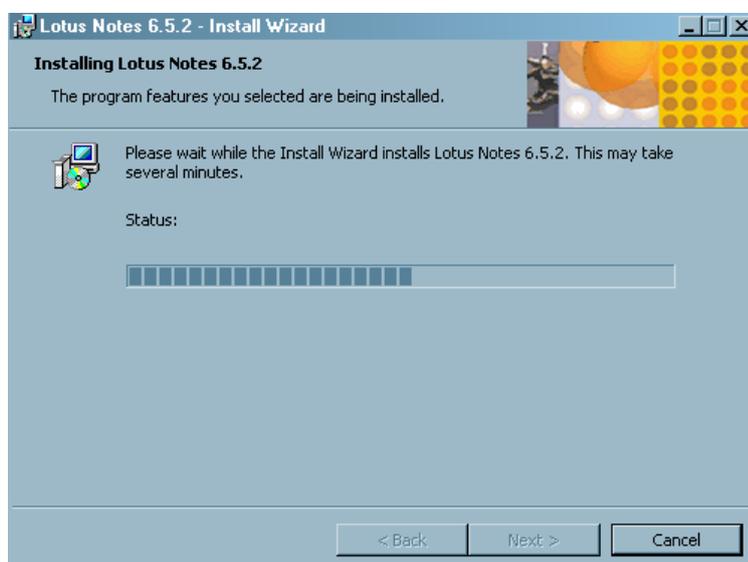


Figura 2.40 Fuente: Instalación de Lotus Notes

12. Haga clic en el botón **Finish**, para finalizar la instalación y proceder con la configuración.

Fin de la instalación de Lotus Notes



Figura 2.41 Fuente: Instalación de Lotus Notes

### 2.1.2.2. Configuración del cliente

1. Levante el servidor, accediendo al menú Inicio, opción Programas, opción Lotus Application, haga clic en **Lotus Domino Server**.

Levantando el Servidor Domino



Figura 2.41 Fuente: Configuración de Lotus Notes

2. Seleccione el servicio con el que va a trabajar el servidor y haga clic en el botón **“Ok”**

Tipo de servicio a levantar

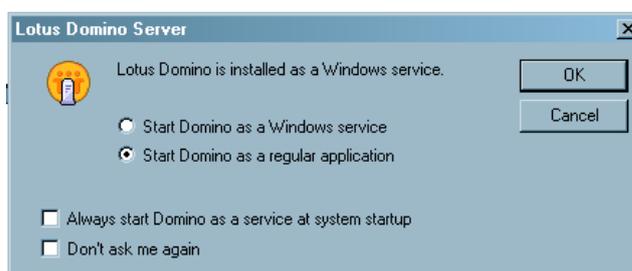


Figura 2.42 Fuente: Configuración de Lotus Notes

### 3. El servidor levanta las tareas con la que fue configurado.

Tareas levantadas en el servidor

```

HUB/AwCorp: Lotus Domino Server
10/10/2006 23:15:09 HTTP Server: Using Web Configuration View
10/10/2006 23:15:10 SMTP Server: Started
10/10/2006 23:15:10 IMAP Server: Started
10/10/2006 23:15:12 SchedMgr: Done validating Schedule Database
10/10/2006 23:15:15 Database Server started
10/10/2006 23:15:15 Platform Stats: Unable to obtain performance data. Please
check Microsoft MSDN on how to enable performance data collection.
10/10/2006 23:15:19 JVM: Java Virtual Machine initialized.
10/10/2006 23:15:19 HTTP Server: Java Virtual Machine loaded
10/10/2006 23:15:23 Setting up default monitors in Monitoring Configuration da
tabase
10/10/2006 23:15:24 HTTP Server: DSAPI Domino Off-Line Services HTTP extension
Loaded successfully
10/10/2006 23:15:26 HTTP Server: Creating Domino Web Administrator...
10/10/2006 23:15:28 LDAP Server: Serving directory names.nsf in the www.corp.c
om Internet domain
10/10/2006 23:15:28 LDAP Schema: Started loading...
10/10/2006 23:15:30 Upgrading the design and data of EVENTS4.NSF...
10/10/2006 23:15:35 LDAP Schema: Finished loading
10/10/2006 23:15:35 LDAP Server: Started verifying directory tree on 'names.n
sf'
10/10/2006 23:15:35 LDAP Server: Finished verifying directory tree on 'names.n
sf'
10/10/2006 23:15:36 LDAP Server: Started
>

```

Figura 2.43 Fuente: Configuración de Lotus Notes

### 4. Abra la herramienta Lotus Notes

Abriendo Lotus Notes

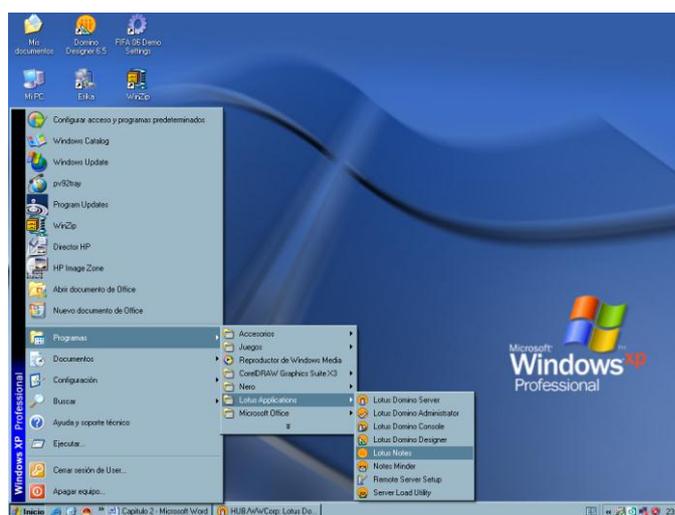


Figura 2.44 Fuente: Configuración de Lotus Notes

## 5. Comienza a cargar la aplicación Lotus Notes

Cargando Lotus Notes

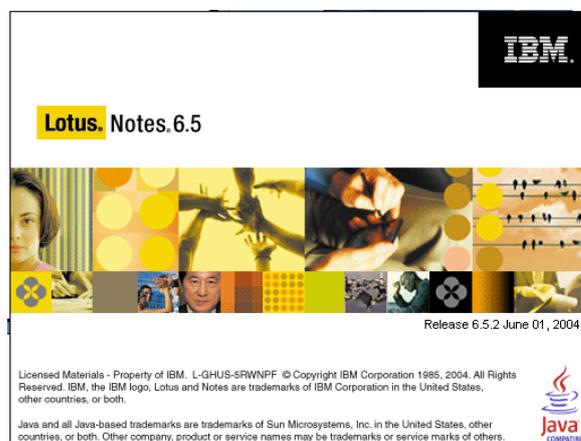


Figura 2.45 Fuente: Configuración de Lotus Notes

## 6. Aparecerá la ventana de bienvenida de la configuración de Lotus Notes, haga clic en el botón “Next”

Bienvenida a la configuración de Lotus Notes

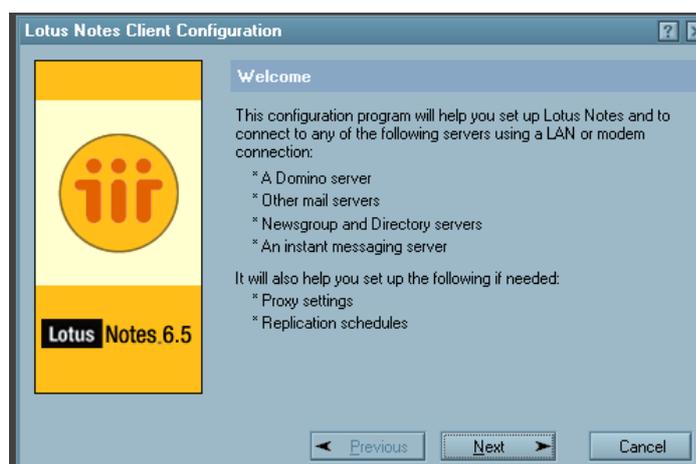


Figura 2.46 Fuente: Configuración de Lotus Notes

7. Ingrese la información acerca de su Nombre y el nombre del servidor, luego haga clic en el botón **“Next”**

Registro de información del usuario y servidor

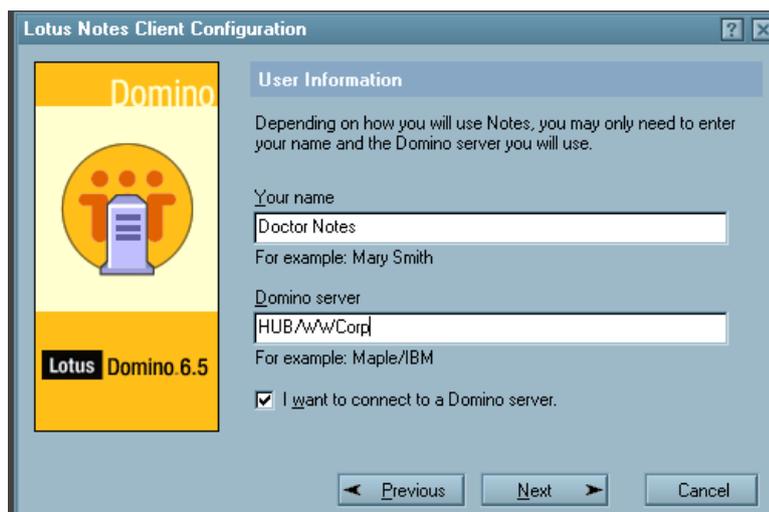


Figura 2.47

Fuente: Configuración de Lotus Notes

8. Ingrese la clave para poder realizar la conexión con el servidor al que este direccionando la configuración, haga clic en el botón **“OK”**

Ingreso de clave para registrar la conexión con el servidor

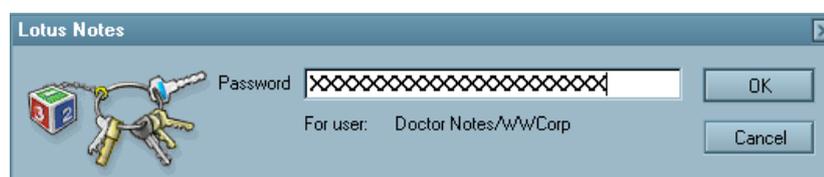


Figura 2.48

Fuente: Configuración de Lotus Notes

9. Haga clic en el botón **“Next”** y saldrá un cuadro de diálogo que me indica que la configuración ha sido exitosa y completa.

Configuración completa

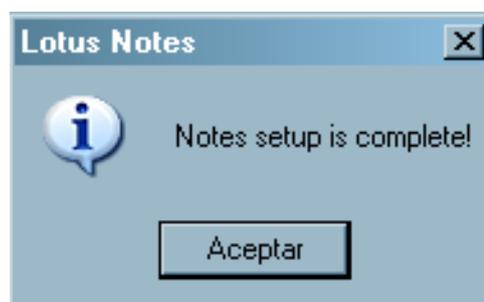


Figura 2.49 Fuente: Configuración de Lotus Notes

10. Finalmente sale la pantalla inicial de Lotus Notes en la cual se puede trabajar.

Ventana de trabajo de Lotus Notes

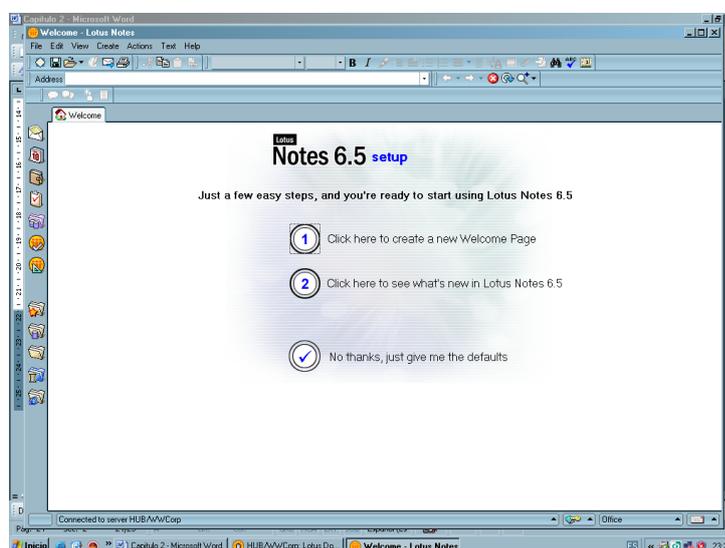


Figura 2.50 Fuente: Configuración de Lotus Notes

### 2.1.3. Configuraciones de Lotus Domino Administrator



1. Levante el servidor y luego haga doble clic en el icono Domino Admin 6.5., para empezar a configurar el documento del servidor.
2. Se abrirá la ventana del Domino Administrator, en la ficha People&Groups debe registrar a los usuarios que van a manejar el workflow. Haciendo clic en la etiqueta **People** ubicado a la derecha de la ventana y luego clic en la opción **Register**.

#### Registro de usuarios

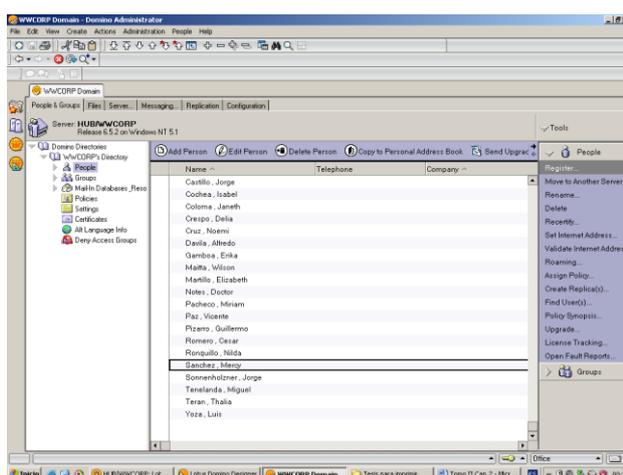


Figura 2.51 Fuente: Configuración del Documento del Servidor

3. Ingrese la clave de administrador para poder crear a nuevos usuarios.

#### Ingreso de clave



Figura 2.52 Fuente: Configuración del documento del Servidor

4. Seleccione el Id del certificador para poder crear los nuevos usuarios, haga clic en el botón **“OK”**

Información de la certificación para la creación de usuarios



Figura 2.53

Fuente: Configuración del documento del Servidor

5. Registre el usuario con los datos requeridos, haga clic en el botón Password Options para sincronizar la contraseña de Internet con la de Notes ID, haga clic en el botón **“OK”** y luego hace clic en el botón que tiene un visto y después en el botón **“Register All”** y por último en el botón **“Done”**.

## Registro de usuarios nuevos

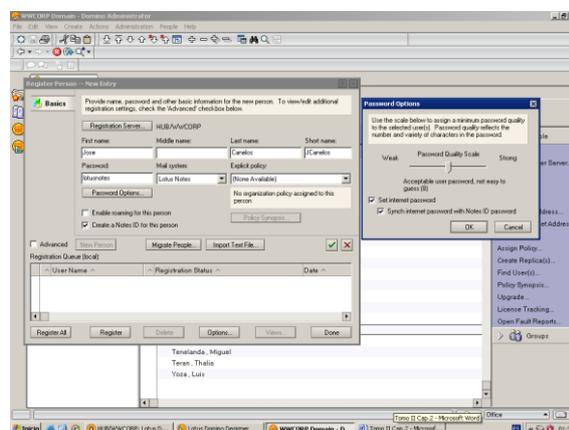


Figura 2.54 Fuente: Configuración del documento del Servidor

- Haga doble clic sobre el nombre del usuario y luego haga doble clic sobre el ícono **UserID** seguido del botón **Save**, ubicando el archivo en una carpeta que debe ser creada sólo para guardar los Ids de cada usuario que va a ser creado.

## Ubicación de los Ids

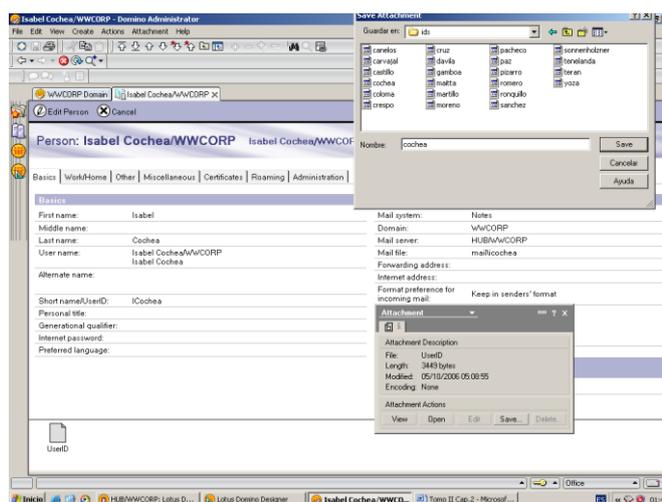


Figura 2.55 Fuente: Configuración del documento del Servidor

7. Luego cree las localidades para cada nuevo usuario, esto permitirá el acceso a cada usuario al workflow.

#### Creación de localidades

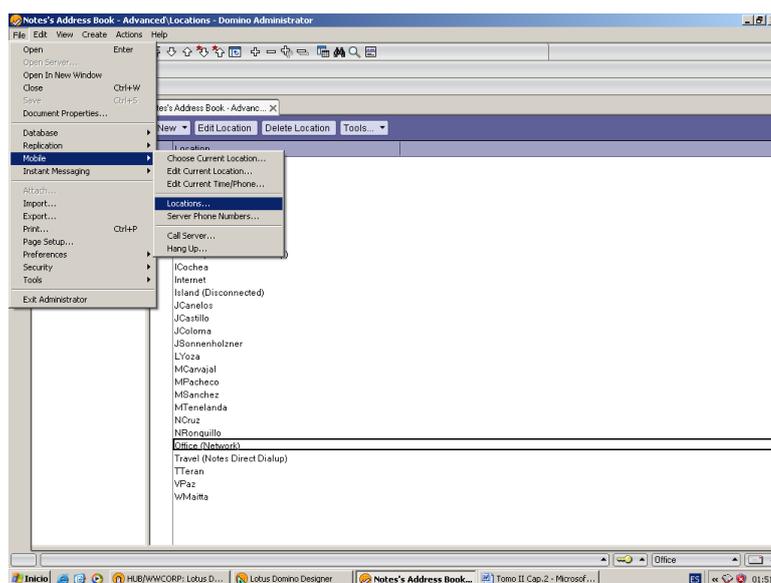


Figura 2.56 Fuente: Configuración del documento del Servidor

8. Haga una copia de Office (NetWork), y edítelo con los datos del nuevo usuario, tales como:
- \* Location Name del Tab Basic
  - \* Mail File del tab Mail; y
  - \* Localice el archivo ID del usuario haciendo clic en el ícono de la lupa del tab Advanced

### Creando un usuario nuevo

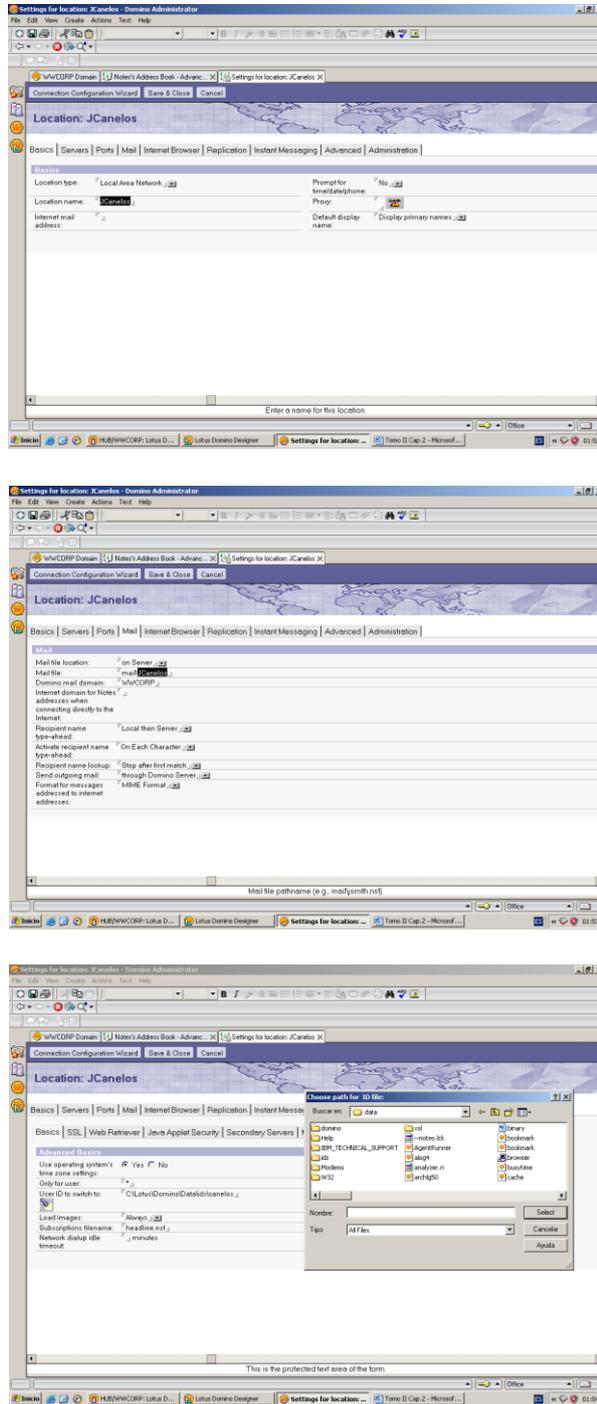


Figura 2.57 Fuente: Configuración del documento del Servidor

9. En el tab Configuration en la menú Server elija la opción **All Server Document** para acceder al servidor que quiere modificar, selecciónelo haciendo doble clic sobre su nombre.

Configurando el servidor

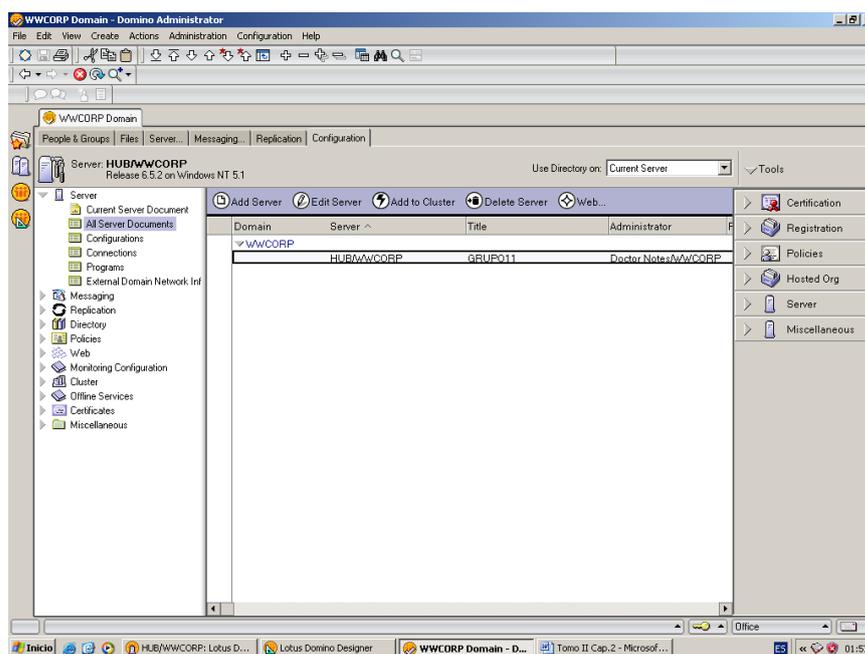


Figura 2.58 Fuente: Configuración del documento del Servidor

10. En el tab Security en la opción **Programmability Restrictions Who can**, añade a los usuarios que podrán correr agentes y códigos java y scripts en la aplicación.

## Configurando las seguridades del servidor

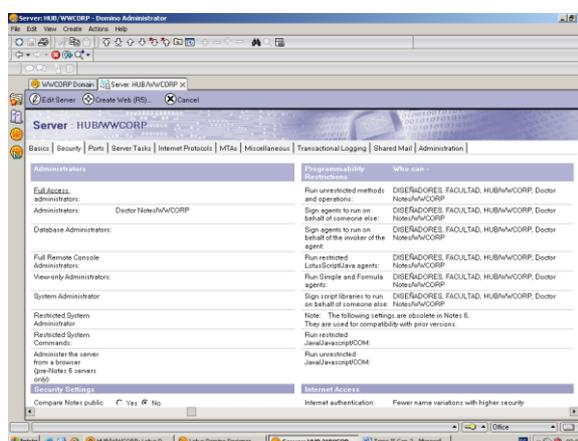


Figura 2.59 Fuente: Configuración de documento del servidor

- En el tab Internet Protocols escoja el tab Domino Web Engine, y en la opción Java Servlets de doble clic para editarlo y seleccione la opción **Domino Servlet Manager**, y luego en la opción **Web Agent** que se encuentra al final de la página seleccione la opción **Enable**. Una vez realizados estos pasos haga clic en el botón **Save&Close**.

## Configurando el tab de Internet

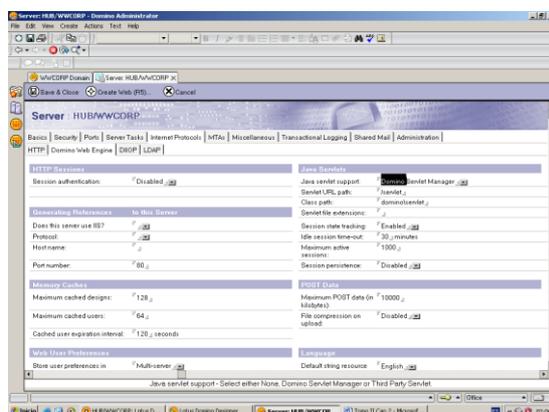


Figura 2.60 Fuente: Configuración del documento del servidor

## 2.1.4. Creación de la base de datos y los elementos de diseño

Para crear la base de datos realice los siguientes pasos:

Ejemplos de bases de datos

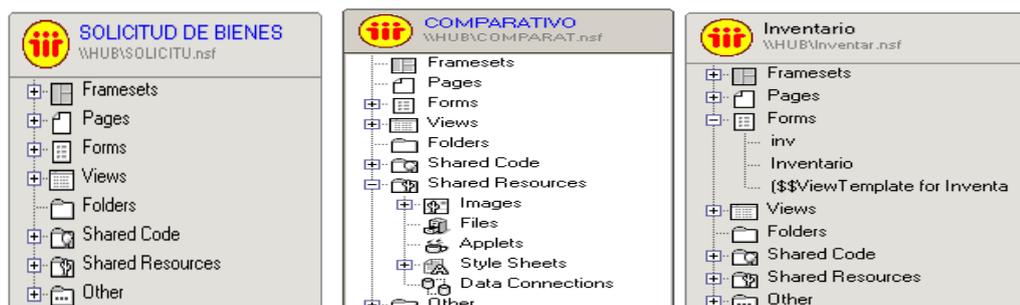


Figura 2.61 Fuente: Lotus Domino Designer 6.5

1. Abra la herramienta Lotus Domino Designer ubicada en el escritorio.
2. Haga clic en el menú File, escoja la opción Database y luego en New.... para así poder crear la nueva base.
3. Luego especifique el nombre y la ubicación de la nueva base de datos
4. Especifique el tipo de plantillas con las que quiere trabajar para la elaboración de alguna aplicación.

## Creación de bases de datos

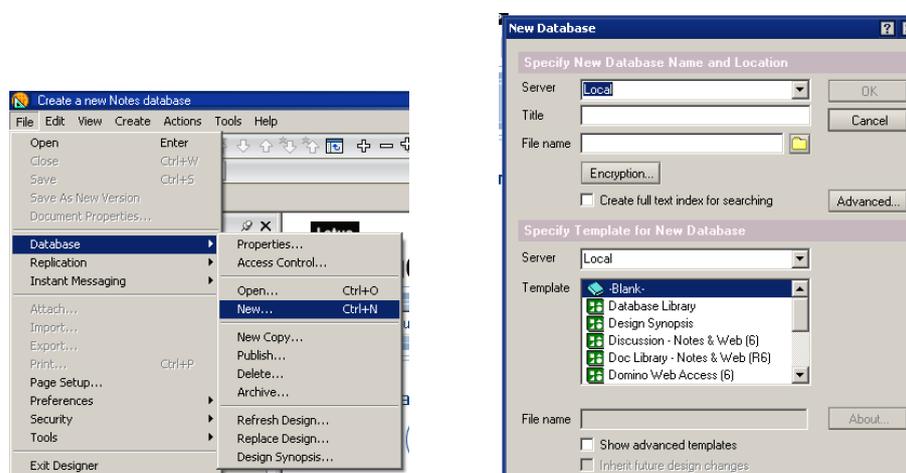


Figura 2.62 Fuente: Lotus Domino Designer 6.5

5. Una vez creada las bases de datos, empezamos a crear los demás elementos de diseño, tales como frameset, páginas, formularios, vistas, agentes, librerías, utilizar imágenes de fondo, etc...

## Base de dato creada

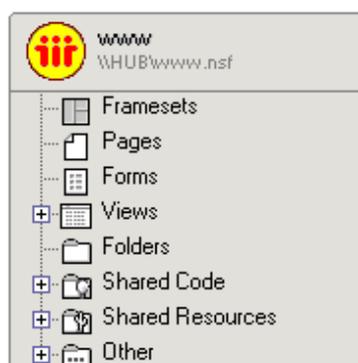


Figura 2.63 Fuente: Lotus Domino Designer 6.5

## **2.2. Manual técnico del Workflow Petición de Bienes**

Para la elaboración del Workflow Petición de Bienes hemos creado 3 bases de datos, con 2 hojas de estilo:

1. SOLICITUD DE BIENES (SOLICITU.nsf), esta base contiene los elementos de diseño para la creación de usuarios, roles, nuevas solicitudes de peticiones de los bienes requeridos por una dependencia de la Facultad de Ciencias Médicas.
2. COMPARATIVO (COMPARAT.nsf), esta base permite la elaboración del cuadro comparativo de cotizaciones y genera una solicitud de compra que contiene la autorización del decano y el administrador. Ya que toda esta documentación es enviada a Administración Central para su aprobación y su compra.
3. Inventario (Inventar.nsf), esta base contiene los elementos para trabajar con los bienes ya adquiridos elaborando un reporte con los códigos de inventarios que son asignados por el jefe de activo fijo de Administración Central y que luego son enviados al administrador y el administrador de la facultad envía al bodeguero

Bases de datos diseñadas para el workflow

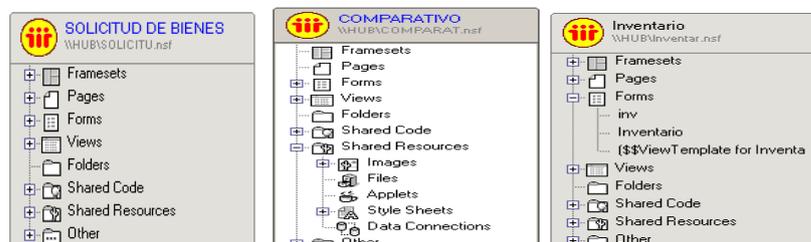


Figura 2.64 Fuente: Lotus Domino Designer 6.5

## 2.2.1. Base SOLICITU.nsf

Esta base de datos fue diseñada para la creación de usuarios, solicitudes de petición, ingreso de bienes, a continuación se describe como fue elaborada la base de datos.

### 2.2.1.1. Creación de frameset

El frameset permite dar una mejor presentación a la aplicación, esto permite adicionar a la aplicación información que debe permanecer constante durante todo el proceso.

Para crear un frameset haga clic sobre la palabra **Frameset** dentro de la base de datos creada y luego haga clic en el botón **New Frameset**. Este puede dividirse en varios paneles según el diseño de su aplicación.

Para el Workflow se elaboró solo un frameset en esta base, llamado **Principal** el cual permite mostrar durante todo el flujo, los menús y el usuario que ha ingresado al Workflow.

Este frameset consta de 3 divisiones y está distribuido de la siguiente manera:

- Sección Izquierda, llamada **izquierda** contiene un elemento del Tipo Name Element: Page y el Value que corresponde al nombre de la página.
- Sección superior, llamada **Principal**, contiene la página Cabecera
- Sección central, llamada **central**, contiene todos los elementos que muestran los resultados de cada proceso dentro del flujo.

Todas estas divisiones son configuradas en las propiedades del frameset. Accedemos a ellas haciendo clic derecho sobre el frameset y escogiendo la opción **Frame Properties**

## Creación de frameset en una aplicación

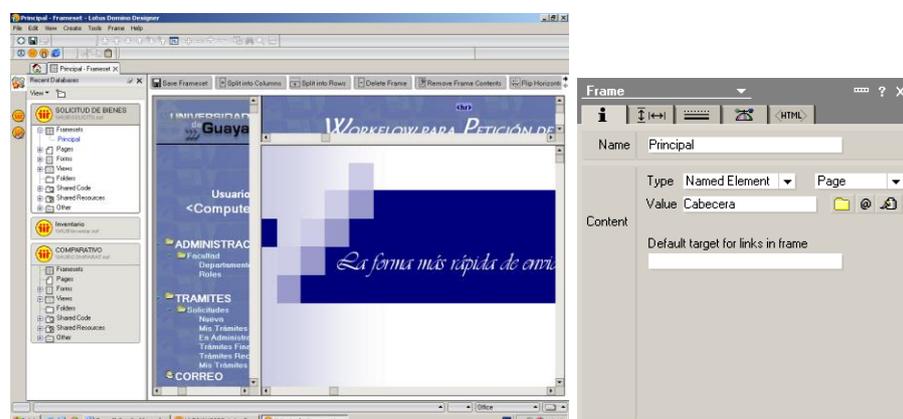


Figura 2.65 Fuente: Lotus Domino Designer 6.5

### 2.2.1.2. Creación de páginas

El diseño de páginas es similar en concepto a una página HTML estática. Las páginas de Domino estructuran y muestran información que permanecen relativamente estable, tales como:

- Información
- Tablas
- Secciones
- Gráficos, y demás elementos de diseño.

Para crear una página haga clic en la palabra Page de la base de datos ya existente y luego haga clic en el botón **New Page**.

Para el workflow de peticiones se crearon 6 páginas, descritas a continuación:

- Cabecera, contiene el título de Workflow **“Workflow Petición de Bienes”**
- Entrada, contiene la página de bienvenida al Workflow. Esta contiene además la codificación script en HTML que muestra en la barra de estado la pagina que se está abriendo.

```

<script language="JavaScript">
var speed = 50
var pause = 3000
var timerID = null
var bannerRunning = false
var ar = new Array()
ar[0] = " Abriendo Workflow de Peticiones....."
ar[1] = ">>>> Universidad de Guayaquil <<<<"
ar[2] = "..... Facutad de Ciencias Médicas ....."
ar[3] = "..... Workflow Petición de Bienes ....."

var currentMessage = 0
var offset = 0
function stopBanner() {
    if (bannerRunning)
        clearTimeout(timerID)
    bannerRunning = false
}
function startBanner() {
    stopBanner()
    showBanner()
}
function showBanner() {
    var text = ar[currentMessage]
    speed = 50
    if (offset < text.length) {
        if (text.charAt(offset) == " ")
            offset++
        var partialMessage = text.substring(0, offset + 1)
    }
}

```

```
        window.status = partialMessage
        offset++ // IE sometimes has trouble with "++offset"
        timerID = setTimeout("showBanner()", speed)
        bannerRunning = true
    } else {
        offset = 0
        currentMessage++
        if (currentMessage == ar.length)
            currentMessage = 1
        timerID = setTimeout("showBanner()", pause)
        bannerRunning = true
    }
}
startBanner();
// -->
</script>
```

- IzquierdaMenu, contiene el menú de opciones al que puede acceder el usuario que ingrese al workflow. Está compuesto por layers, y outlines.

Para crear un elemento de diseño debemos ir al menú **Create** y encontraremos los elementos que deseemos crear y adicionar, tales como layers, campos, botones, etc...

Menú Create con la lista de opciones

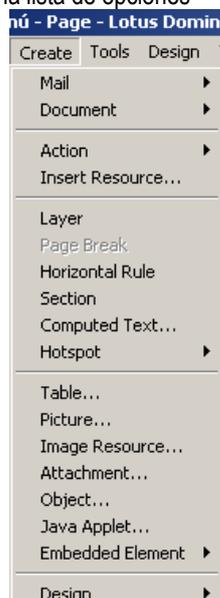


Figura 2.66 Fuente: Lotus Domino Designer

- `Popup_calendar`, contiene la siguiente codificación que sirve para realizar un botón de calendario cada vez que este es invocado. Para que este funcione debe activarse en el menú Text la opción **Pass Thru HTML**

```

<HTML>

<head>
<title>calendar</title>
<Style>
a:visited {
    color: #000000;
    text-decoration : none;
}

a:link,active {
    color: #000000;
    text-decoration : none;
}

.daytype {font-family : Arial; font-weight:bold; font-size : 9pt;}

```

```

.otherday {font-family : Arial; font-weight:normal; color:#808080; font-size :
9pt; background-color:#FFFFFF; }
.thisday {font-family : Arial; font-weight:normal; font-size : 9pt; background-
color:#FFFFFF; }
.todate {font-family : Arial; font-weight:normal; font-size : 9pt; background-
color:#FFFFFF; }
.sunday {font-family : Arial; font-weight:normal; font-size : 9pt; background-
color:#FFFFFF; }
</Style>

```

```

<script type="text/javascript" language="JavaScript">
<!--

var names = new array("1","2","3","4","5","6","7","8","9","10","11","12");
var days = new array(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
var dow = new array("D","L","M","M","J","V","S","","","");

var today = new Date();
var thisDay = today.getDate();
var thisMonth = today.getMonth();
var thisYear = today.getFullYear();

var args = window.dialogArguments;
var YEAR = ""; var MONTH = "";

function init() {
YEAR = args[0];
MONTH = args[1];

document.all.calendar_body.innerHTML = Calendar(MONTH,YEAR);
}

function getParameter (strValue, parameterName, sep) {
var return_value = "";
var lists = strValue.split(sep);
var lists2 = new Array();

for (var i=0; i < lists.length; i++) {
lists2 = lists[i].split("=");
if (lists2[0] == parameterName) { return_value = lists2[1]; break; }
}

return return_value;
}

function Calendar(Month,Year)
{
var output = "";
output += "<form name='Cal'>";
output += "<table bgcolor='white' width='150' align=center><tr>";

```

```

output += "<td width=\"10%\" height='15' align=left noresize nowrap>";
output += "<span style=\"cursor:hand;\" onclick=\"prev_month()\"><IMG
SRC=\"arrw04d.gif?OpenImageResource\" border=0></span>";
output += "</TD>";
output += "<td width=\"80%\" align=center noresize nowrap>";
output += "<font size=-1>";
output += "<select name='Year' onChange='changeYear();'>";
for (year=1950; year<2051; year++)
{
    if (year == YEAR)
        output += "<option value=\"" + year + "\" selected>" +
year + "</option>";
    else
        output += "<option value=\"" + year + "\">" + year +
"</option>";
}
output += "</select>";
output += "/";
output += "<select name='Month' onChange='changeMonth();'>";
for (month=0; month<12; month++)
{
    if (month == MONTH)
        output += "<option value=\"" + month + "\" selected>" +
names[month] + "</option>";
    else
        output += "<option value=\"" + month + "\">" +
names[month] + "</option>";
}
output += "</select>";
output += "</td>";
output += "</font>";

output += "<td width=\"10%\" align=right noresize nowrap>";
output += "<span style=\"cursor:hand;\"
onclick=\"next_month();\"><IMG
SRC=\"arrw04e.gif?OpenImageResource\" border=0></span>";
output += "</TD>";
output += "</TR></TABLE>";

firstDay = new Date(Year,Month,1);
startDay = firstDay.getDay();

if (((Year % 4 == 0) && (Year % 100 != 0)) || (Year % 400 == 0))
    days[1] = 29;
else
    days[1] = 28;

output += "<table cellspacing='0' cellpadding='0' border=0
bgcolor='#D0DFEF' width=\"167\" align=center>";
output += "<tr><td>";

```

```

output += "<table cellspacing='1' cellpadding='1' border=0
width='167'\>";
output += "<tr><td>";

output += "<table cellspacing=0 cellpadding=0 border=0
width='167'\>";
output += "<tr>";
for (i=0; i<7; i++)
{
    output += "<td width=20 height=20 align=center
valign=middle bgcolor='#E7EFF7' class='daytype' noresize nowrap>";

    output += "<B>" + dow[i] + "</td><\b>";
}
output += "</tr>";
output += "<tr align=center valign=middle>";
var column = 0;
var lastMonth = Month - 1;
if (lastMonth == -1)
lastMonth = 11;
for (i=0; i<startDay; i++)
{
    output += "<td width=20 height=20 class='otherday' noresize
nowrap>"
    output += (days[lastMonth]-startDay+i+1) + "</td>";
    column++;
}

var date1;
for (i=1; i<=days[Month]; i++)
{
    date1 = new Date(Year, parseInt(Month), i);

    if (date1.getDay() == 0) {
        output += "<td width=20 height=20 class='sunday'
noresize nowrap>" + "<span style='cursor:hand;\
onclick='\"set_date(\"+Year+\", \"+(parseInt(Month)+1)+\", \"+i+\"')\">"+
"<font style='font-weight:bold;color=red'>"+i + "</font></span></td>";
    } else {
        if ((i == thisDay) && (Month == thisMonth) && (Year
== thisYear))
            output += "<td width=20 height=20
class='todate' noresize nowrap>" + "<span style='cursor:hand;\
onclick='\"set_date(\"+Year+\", \"+(parseInt(Month)+1)+\", \"+i+\"')\">"+
"<font style='font-weight:bold;color=blue'>"+i +
"</font></span></td>";
        else
            output += "<td width=20 height=20
class='thisday' noresize nowrap>" + "<span style='cursor:hand;\
onclick='\"set_date(\"+Year+\", \"+(parseInt(Month)+1)+\", \"+i+\"')\">"+ i
+ "</span></td>";
    }
}

```

```

    }

    column++;
    if (column == 7)
    {
        output += "<\tr><tr align=center valign=middle>";
        column = 0;
    }
}
if (column > 0)
{
    for (i=1; column<7; i++)
    {
        output += "<td width=20 height=20 class='otherday'
noresize nowrap>" + i + "<\td>";
        column++;
    }
}

output += "</td><\tr></table>";
output += "</td><\tr></table>";
output += "</td></tr></table><\form>";

return output;
}

function array(m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11)
{
    this[0] = m0; this[1] = m1; this[2] = m2; this[3] = m3;
    this[4] = m4; this[5] = m5; this[6] = m6; this[7] = m7;
    this[8] = m8; this[9] = m9; this[10] = m10; this[11] = m11;
}

function changeMonth () {
    if
    (document.Cal.Month.options[document.Cal.Month.selectedIndex].va
lue != "")
    {
        YEAR =
document.Cal.Year.options[document.Cal.Year.selectedIndex].value;
        MONTH =
document.Cal.Month.options[document.Cal.Month.selectedIndex].val
ue;

        document.all.calendar_body.innerHTML =
Calendar(MONTH, YEAR);
    }
}

function prev_month() {
    //if first year and first month then return

```

```

if (document.Cal.Year.selectedIndex == 0 &&
document.Cal.Month.selectedIndex == 0) return;

YEAR = (MONTH == 0) ? YEAR - 1 : YEAR;
MONTH = (MONTH == 0) ? 11 : MONTH - 1;

document.all.calendar_body.innerHTML = Calendar(MONTH, YEAR);
}

function next_month() {
//if last year and last month then return
if (document.Cal.Year.selectedIndex == document.Cal.Year.length -
1 &&
document.Cal.Month.selectedIndex == document.Cal.Month.length -
1) return;

YEAR = (MONTH == 11) ? eval(YEAR) + 1 : YEAR;
MONTH = (MONTH == 11) ? 0 : eval(MONTH) + 1;

document.all.calendar_body.innerHTML = Calendar(MONTH, YEAR);
}

function changeYear () {
if
(document.Cal.Year.options[document.Cal.Year.selectedIndex].value
!= "")
{
YEAR =
document.Cal.Year.options[document.Cal.Year.selectedIndex].value;
MONTH =
document.Cal.Month.options[document.Cal.Month.selectedIndex].val
ue;

document.all.calendar_body.innerHTML =
Calendar(MONTH, YEAR);
}
}

function y2k(number) { return (number < 1000) ? number + 1900 :
number; }

function set_date(Y, M, D) {
window.returnValue = make_two_digit(D)+"/" +
make_two_digit(M)+"/" + make_two_digit_year(Y);
window.close();
}
function make_two_digit(x) {
if (x.toString().length == 1) return "0"+x; else return x;
}
function make_two_digit_year(x) {

```

```

if (x.toString().length == 4) return x.toString().substring(2,4); else
return x;
}
//-->
</script>
</head>

<body bgcolor="white" marginwidth=1 marginheight=0 topmargin=0
leftmargin=0 onload="init();">
<p>
<div id="calendar_body" style="margin-left:10px;"></div>

</body>
</html>

```

- Salida contiene un layer que me indica que el proceso se realizó de manera exitosa.

#### Creación de páginas domino

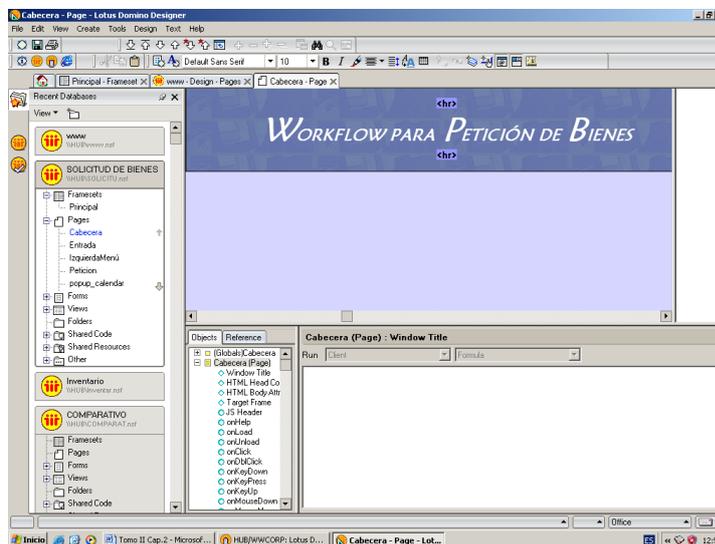


Figura 2.67 Fuente: Lotus Domino Designer

### 2.2.1.3. Creación de forms

Los formularios son elementos de diseño que permiten introducir datos. Estos contienen la lógica de la aplicación y pueden realizar acciones sobre los datos.

Para crear un nuevo formulario sólo debemos hacer clic en la palabra Forms y luego clic en el botón **New Forms**.

Para nuestra aplicación hemos creado 6 formularios básicos y 10 formularios basados en una plantilla existente.

- Bienes
- Departamentos
- Perfil
- Roles
- Solicitud de Compra
- Solicitudes
- \$\$View Template for default
- \$\$View Template for Departamentos
- \$\$View Template for MailUsuarios
- \$\$View Template for Mis Solicitudes
- \$\$View Template for Solicitud de Compra
- \$\$View Template for Solicitudes en Administración Central

- \$\$View Template for Solicitudes en Proceso Facultad
- \$\$View Template for Solicitudes
- \$\$View Template for Trámites Finalizados
- \$\$View Template for Trámites Rechazados

#### Creación de formularios

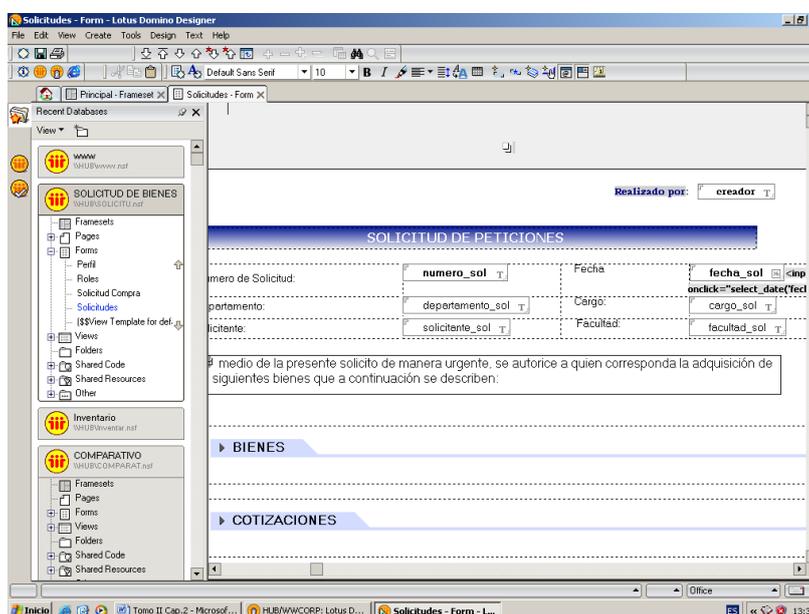


Figura 2.68 Fuente: Lotus Domino Designer

Cada uno de ellos contiene campos, vistas, secciones, y acciones que permiten controlar el proceso del flujo del workflow.

El valor de los campos son modificados en el **default value** en la ficha de objetos del panel de programación, o accediendo a las propiedades del campo haciendo clic derecho sobre el campo y escogiendo la opción **Field Properties**.

En todas las acciones de Imprimir se utilizó el siguiente código Script.

```
<SCRIPT LANGUAGE='JavaScript'>
<!--
function imprimir() {
  version = parseInt(navigator.appVersion);
  if (version >= 4)
    window.print();
}
// --></SCRIPT>
```

Además se hace uso del siguiente código HTML para personalizar la barra de estado.

```
<script language="JavaScript">
var speed = 50
var pause = 3000
var timerID = null
var bannerRunning = false
var ar = new Array()

ar[0] = " Abriendo XXXXXXXXXXXXXXXXXXXX....."

ar[1] = ">>>> Universidad de Guayaquil <<<<"
ar[2] = "..... Facutad de Ciencias Médicas ....."
ar[3] = "..... Workflow Petición de Bienes ....."

var currentMessage = 0
var offset = 0
function stopBanner() {
  if (bannerRunning)
    clearTimeout(timerID)
  bannerRunning = false
}

function startBanner() {
  stopBanner()
  showBanner()
```

```

}

function showBanner() {
var text = ar[currentMessage]
speed = 50
if (offset < text.length) {
    if (text.charAt(offset) == " ")
        offset++
    var partialMessage = text.substring(0, offset + 1)
    window.status = partialMessage
    offset++ // IE sometimes has trouble with "++offset"
    timerID = setTimeout("showBanner()", speed)
    bannerRunning = true
} else {
    offset = 0
    currentMessage++
    if (currentMessage == ar.length)
        currentMessage = 1
    timerID = setTimeout("showBanner()", pause)
    bannerRunning = true
}
}

startBanner();

// -->
</script>

```

A continuación describimos el código para cada formulario

- **Formulario Bienes**

### Campos usados

Campo	Tipo de Campo	Valor del Campo
<b>Num_solicitud</b>	Text	Uso de lenguaje de fórmula <b>@UrlQueryString("num_solicitud")</b>
<b>Fecha_bien</b>	Data Time	Uso de lenguaje de fórmula <b>@Created</b>
<b>Tipo_bien</b>	Dialog List	Lista de opciones
<b>Desc_bien</b>	Text	Ingreso de información
<b>Cantidad</b>	Number	Ingreso de información
<b>Id_sol</b>	Text	Campo heredado Default Value = id
<b>Id_sol_1</b>	Text	Uso de lenguaje de fórmula <b>@Text(@DocumentUniqueId)</b>

Tabla 2.1

Fuente: Grupo de Proyecto

## Acciones

Accion	Código
<b>Nuevo</b>	Uso de lenguaje de fórmula <b>var:= num_solicitud;</b> <b>@URLOpen("http://hub/SOLICITU.nsf/Bienes?OpenForm&amp;num_solicitud="+var)</b>
<b>Guardar</b>	Uso de lenguaje de fórmula <b>@Command([FileSave])</b>
<b>Editar</b>	Uso de lenguaje de fórmula <b>@Command([EditDocument])</b>
<b>Regresar</b>	Uso de lenguaje de fórmula <b>num_sol:=@Text(num_solicitud);</b> <b>@Command([OpenView];"Solicitudes";num_sol);</b> <b>@Command([OpenDocument])</b>

Tabla 2.2

Fuente: Grupo de Proyecto

- **Formulario Departamentos**

## Campos usados

Campo	Tipo de Campo	Valor del Campo
<b>Facultad_dep</b>	Text	Ingreso de información
<b>Escuela_dep</b>	Text	Ingreso de información
<b>Descripción_dep</b>	Text	Ingreso de información
<b>Cargo_dep</b>	Text	Ingreso de información
<b>Solicitante_dep</b>	Text	Ingreso de información

Tabla 2.3

Fuente: Grupo de Proyecto

## Acciones

Accion	Código
<b>Nuevo</b>	Uso de lenguaje de fórmula <b>@Command([Compose];"Departamentos")</b>
<b>Guardar</b>	Uso de lenguaje de fórmula <b>@Command([FileSave])</b>
<b>Salir</b>	Uso de lenguaje de fórmula <b>@SetTargetFrame("central");</b> <b>@URLOpen("http://Hub/SOLICITU.nsf/Salida?OpenPage")</b>

Tabla 2.4

Fuente: Grupo de Proyecto

- **Formulario Roles**

### Campos usados

Campo	Tipo de Campo	Valor del Campo
Facultad_dep	Text	Ingreso de información
Departamento_dep	Text	Ingreso de información
Nombre_dep	Text	Ingreso de información
Apellido_dep	Text	Ingreso de información
Apellidoma_dep	Text	Ingreso de información
Descripción_dep	Text	Ingreso de información

Tabla 2.7

Fuente: Grupo de Proyecto

### Acciones

Accion	Código
<b>Nuevo</b>	Uso de lenguaje de fórmula <b>@Command([Compose];"Roles")</b>
<b>Guardar</b>	Uso de lenguaje de fórmula <b>nombre:=nombres_rol;</b> <b>apellido:=apellido_rol;</b> <b>nombre_completo:=nombres_rol+" "+apellido_rol;</b> <b>@SetField("mail_us";@Left(nombre; 1)+@Left(@Word(apellido ; " "; 1);7)+".nsf");</b> <b>html:= "http://hub/mail/";</b> <b>@SetField("ruta_mail";html+mail_us);</b> <b>@SetField("nombre_rol";@Word( nombre ; " "; 1)+</b> <b>" +apellido_rol);</b> <b>@Command([FileSave])</b>
<b>Salir</b>	Uso de lenguaje de fórmula <b>@SetTargetFrame("central");</b> <b>@URLOpen("http://Hub/SOLICITU.nsf/Salida?OpenPage")</b>

Tabla 2.8

Fuente: Grupo de Proyecto

- **Formulario Solicitud de Compra**

### Campos usados

Campo	Tipo de Campo	Valor del Campo Uso de Lenguaje de Fórmula
creador	Text	usuario:=@Name([CN];@UserName); @Text(usuario); Usuario
compra_sol	Text	@Text(numero_sol)
Fecha compra	Date/Time	@Created
Solicitante compra	Text	@Text(solicitante_sol)
Cargo compra	Text	@Text(cargo_sol)
Departamento compra	Text	@Text(departamento_sol)
Facultad compra	Text	@Text(facultad_sol)
<b>Campos de Sección</b>		
<u>Sumilla</u>	Text	Ingreso de información
Justificativo	Text	Ingreso de información
Pedido por	Text	Ingreso de información
Tiempo	Text	Ingreso de información
Observaciones		

Tabla 2.9

Fuente: Grupo de Proyecto

### Acciones

Accion	Código
Grabar	Uso de lenguaje de fórmula FIELD estado:="Revision Jefe Adquisicion" ; @Command([FileSave]); @True
Regresar	Uso de lenguaje de fórmula num_sol:=@Text(compra_sol); @Command([OpenView];"Solicitudes";num_sol); @Command([OpenDocument])
Enviar	Uso de lenguaje de fórmula FIELD boton:=6; @SetField("fecha_aprobacion";@Now); @Command([FileSave]); @Command([ToolsRunMacro]; "(Avanzar)")
Codificar Bienes	Uso de lenguaje de fórmula num:= compra_sol; depa:=departamento_compra; facu:=facultad_compra; solicitante:=solicitante_compra; FIELD estado:="Finalizado"; @URLOpen("http://HUB/Inventar.nsf/Inventario?OpenForm&numoficio="+num+"&area_solicitante="+depa+"&facu_solicitante="+facu+"&solicitante_pet="+solicitante)
Imprimir	Uso de código Java Script Imprimir()

Tabla 2.10

Fuente: Grupo de Proyecto

## Secciones y Vistas Embebidas

Seccion	Vista	Código
Bienes	Bienes	<b>Embedded Selection</b>
Sumilla	Campos	<b>Show Single Category:</b> compra_sol Ingreso de Información

Tabla 2.11

Fuente: Grupo de Proyecto

- **Formulario Solicitudes**

### Campos usados

Campo	Tipo de Campo	Valor del Campo Uso de Lenguaje de Fórmula
creador	Text	<b>usuario:=@Name([CN];@UserName);</b> <b>@Text(usuario);</b> <b>Usuario</b>
numero_sol Fecha_compra	Text Date/Time	Ingreso de información <b>@Created</b> <b>&lt;input type=button value="&gt;&gt;"</b> <b>onclick="select_date('fecha');"&gt;</b>
departamento_sol	Text	<b>departamento_sol:=@DbLookup("":</b> <b>"";"Rol";creador;3);</b> <b>@Text(@If(@IsError(departamento_s</b> <b>ol);"Error";departamento_sol))</b>
Cargo_sol	Text	<b>cargo_sol:=@DbLookup("":</b> <b>"";"Departamentos";departamento_sol;5</b> <b>);</b> <b>@Text(@If(@IsError(cargo_sol);"</b> <b>ca</b> <b>rgo_sol))</b>
solicitante_sol	Text	<b>solicitante_sol:=@DbLookup("":</b> <b>"";"Departamentos";departamento_sol;2);</b> <b>@Text(@If(@IsError(solicitante_sol);</b> <b>"Error";solicitante_sol))</b>
Facultad_sol	Text	<b>facultad_sol:=@DbLookup("":</b> <b>"";"Departamentos";departamento_sol</b> <b>;3);</b> <b>@Text(@If(@IsError(facultad_sol);"</b> <b>facultad_sol))</b>
<b><u>Campos de Sección</u></b> <b><u>Instrucciones de M.</u></b> <b><u>Modificacion_sol</u></b>	Rich Text	Ingreso de información

Tabla 2.12

Fuente: Grupo de Proyecto

## Acciones

Accion	Código
Nueva	Uso de lenguaje de fórmula <b>@Command([Compose];"Solicitudes")</b>
Cotizaciones	Uso de lenguaje de fórmula <b>var:= numero_sol;</b> <b>sol:=departamento_sol;</b> <b>@URLOpen("http://hub/COMPARAT.nsf/Solicitudes_Comparativo?OpenForm&amp;numoficio="+var+"&amp;area_solicitante="+sol)</b>
Ver Cotizaciones	Uso de lenguaje de fórmula <b>num_sol:=@Text(numero_sol);</b> <b>@URLOpen("http://hub/COMPARAT.nsf/Comparativo/"+num_sol+"?OpenDocument")</b>
Editar	Uso de lenguaje de fórmula <b>@Command([EditDocument])</b>
Guardar	Uso de lenguaje de fórmula <b>FIELD boton:=0;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Devolver	Uso de lenguaje de fórmula <b>FIELD boton:=8;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Denegar	Uso de lenguaje de fórmula <b>FIELD boton:=7;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Enviar	Uso de lenguaje de fórmula <b>FIELD boton:=1;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Ingresar bien	Uso de lenguaje de fórmula <b>var:= numero_sol;</b> <b>@URLOpen("http://hub/SOLICITU.nsf/Bienes?OpenForm&amp;num_solicitud="+var)</b>
Enviar	Uso de lenguaje de fórmula <b>FIELD boton:=2;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@SetField("fecha_envio";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Generar Sol. de Compra	Uso de lenguaje de fórmula <b>@Command([Compose];"Solicitud Compra")</b>
Enviar	Uso de lenguaje de fórmula <b>FIELD boton:=3;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@SetField("fecha_aprobacion_1";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
Enviar	Uso de lenguaje de fórmula <b>FIELD boton:=4;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@SetField("fecha_aprobacion_1";@Now);</b>

<b>Enviar</b>	<b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b> Uso de lenguaje de fórmula <b>FIELD boton:=5;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@SetField("fecha_aprobacion_1";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
<b>Denegar</b>	Uso de lenguaje de fórmula <b>FIELD boton:=9;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
<b>Ver Sol. Compra</b>	Uso de lenguaje de fórmula <b>num_sol:=@Text(numero_sol);</b> <b>@URLOpen("http://hub/SOLICITU.nsf/Solicitud Compra 2/"+num_sol+"?OpenDocument")</b>
<b>Imprimir Denegar</b>	<b>Imprimir()</b> Uso de lenguaje de fórmula <b>FIELD boton:=10;</b> <b>@SetField("fecha_aprobacion";@Now);</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Avanzar)")</b>
<b>Salir</b>	Uso de lenguaje de fórmula <b>@SetTargetFrame("central");</b> <b>@URLOpen("http://Hub/SOLICITU.nsf/Salida?OpenPage")</b>

Tabla 2.13

Fuente: Grupo de Proyecto

## Secciones y Vistas Embebidas

Seccion	Vista	Código
<b>Bienes</b>	Bienes	<b>Embedded Selection</b>
<b>Cotizaciones</b>	Comparativo	<b>Show Single Category:</b> numero_sol
<b>Instrucciones de modificacion</b>	Campos	<b>Show Single Category:</b> numero_sol Ingreso de Información

Tabla 2.14

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for default**

## Vistas Embebidas

Vista	Código
Solicitudes Rechazadas	<b>Embedded Selection</b>

Tabla 2.15

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Departamentos**

### Vistas Embebidas

Vista	Código
Departamentos	<b>Embedded Selection</b>

Tabla 2.16

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for MailUsuarios**

### Vistas Embebidas

Vista	Código
Mail Usuarios	<b>Embedded Selection</b> <b>Show single category: usuario</b>

Tabla 2.17

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Mis Solicitudes**

### Vistas Embebidas

Vista	Código
Mis Solicitudes	<b>Embedded Selection</b> <b>Show Single Category: usuario</b>

Tabla 2.18

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Solicitud de Compra**

### Vistas Embebidas

Vista	Código
Solicitud de Compra	<b>Embedded Selection</b> <b>Show Single Category: usuario</b>

Tabla 2.19

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Solicitudes en Admin. Central**

#### **Vistas Embebidas**

<b>Vista</b>	<b>Código</b>
Solicitudes en Administración	<b>Embedded Selection</b> <b>Show Single Category: usuario</b>

Tabla 2.20

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Solicitudes en Proceso Facultad**

#### **Vistas Embebidas**

<b>Vista</b>	<b>Código</b>
Solicitudes en Proceso Facultad	<b>Embedded Selection</b> <b>Show Single Category: usuario</b>

Tabla 2.21

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Solicitudes**

#### **Vistas Embebidas**

<b>Vista</b>	<b>Código</b>
Solicitudes	<b>Embedded Selection</b>

Tabla 2.22

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Trámites Finalizados**

#### Vistas Embebidas

Vista	Código
Trámites Finalizados	<b>Embedded Selection</b>

Tabla 2.23

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Trámites Rechazados**

#### Vistas Embebidas

Vista	Código
Solicitudes Rechazadas	<b>Embedded Selection</b> <b>Show Single Category:</b> usuario

Tabla 2.24

Fuente: Grupo de Proyecto

#### 2.2.1.4. Creación de view

La finalidad de una vista es crear una lista organizada de documentos para que los usuarios puedan localizar la información que necesitan de manera inmediata y precisa.

Para crear una vista haga clic en la palabra view de la base de datos y luego haga clic en el botón **New Vista**

Para nuestro workflow hemos creado 12 vistas y 1 que se crea por default en ella constan todos los documentos que han sido creados.

Cada vista esta relacionada con un formulario lo cual se codifica en el View Selection de la sección objetos del panel de programación.

Vista de Solicitudes creadas

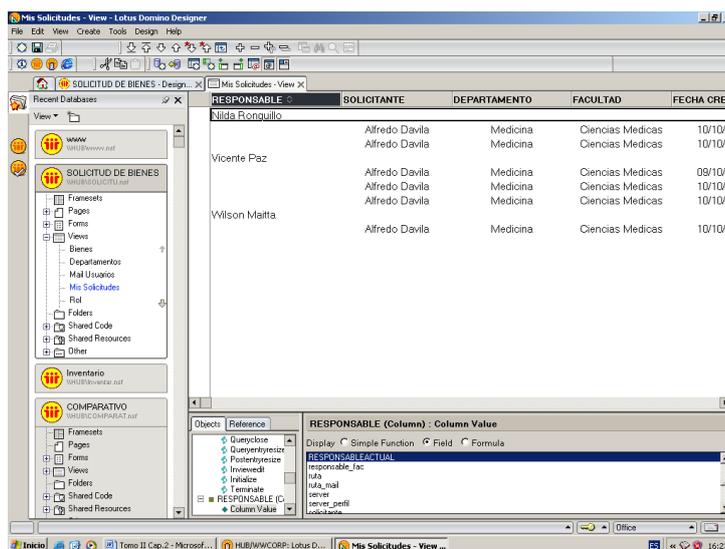


Figura 2.69 Fuente: Lotus Domino Designer 6.5

Vistas creadas

Vista	View Selection	Campos-Columnas
Bienes	SELECT form = "BIENES"	Num_Solicitud (Organizado en forma ascendente y categorizado)
Departamentos	SELECT form = "Departamentos"	Cada columna esta relacionada con un campo
Mail Usuario	SELECT form ="Roles"	Usuario (Organizado en forma ascendente y categorizado)
Mis Solicitudes	SELECT form = "Solicitudes" & estado != "Solicitud Rechazada"	Responsable (Organizado en forma ascendente y categorizado)
Rol	SELECT form ="Roles"	Cada columna esta relacionada con un campo
Solicitud de Compra	SELECT form ="Solicitud Compra"	Responsable (Organizado en forma ascendente y categorizado)
Solicitud de Compra1	SELECT form ="Solicitud Compra"	Cada columna esta relacionada con un campo

Solicitudes	SELECT form ="Solicitudes"	Num_Solicitud (Organizado en forma ascendente y categorizado)
Solicitudes en Administración	SELECT form="Solicitudes" & ( Estado="Revision Vicerrector Administrativo"   Estado="Revision Jefe Adquisicion"   Estado="Revision Jefe Activo" )	Estado (Organizado en forma ascendente y categorizado) <b>Formula:</b> @If(Estado="", "Revision Administrador"; Estado)
Solicitudes en Proceso Facultad	SELECT form="Solicitudes" & ( Estado="Por Enviar"   Estado="Revision Director"   Estado = "Revision Decano"   Estado = "Revision Administrador" )	Estado (Organizado en forma ascendente y categorizado) <b>Formula:</b> @If(Estado="", "Por Enviar"; Estado)
Solicitudes Rechazadas	SELECT form = "Solicitudes" & estado ="Solicitud Rechazada"	Cada columna esta relacionada con un campo

Tabla 2.25 Fuente: Grupo de Proyecto

### 2.2.1.1. Creación de shared code

En este workflow se hace uso de agentes, outilines y Script libraries, los **agentes** son programas que realizan una serie de tareas para que sean ejecutadas por el usuario o de manera automática en una aplicación Domino.

Para crear un agente haga clic sobre la palabra Shared Code, luego en la palabra Agent y finalmente en la opción **New Agent**

Estos agentes pueden ser codificados en Lenguaje de Fórmula, Simple Action, javaScript, LousScript.

Se realizò 1 Agente:

1. **Avanzar**, permite avanzar un proceso dentro del flujo es llamado mediante acciones y el código se describe a continuación:

#### **Options**

Option Public

Option Declare

Use "Correo"

#### **Declarations**

Dim sesion As NotesSession

Dim doc As NotesDocument

Dim db As NotesDatabase

Dim DocMens As NotesDocument

Dim logitem As Variant

#### **Initialize**

Sub Initialize

Dim estado\_actual As Variant

Dim logitem As Variant

Dim responsable, responsableactual As Variant

Dim director, decano, viceadministrativo, jefeadqui As Variant

Dim jefeactivo, administrador, jefebod As Variant

Dim usuario As Variant

Dim creador, docid As Variant

```
Dim nombre As String
Dim view As NotesView
Dim boton As Variant
Dim server, base1 As Variant
Set sesion = New NotesSession
Set doc = sesion.DocumentContext
Dim docvista, docvista1, docvista2, docvista3, docvista4, docvista5,
docvista6 As Notesdocument
Dim item As NotesItem

Set db = sesion.currentdatabase

docid = doc.GetItemValue("id")
estado_actual=doc.GetItemValue("estado")
boton=doc.GetItemValue("boton")
usuario =doc.GetItemValue("resp_actual")
'solicitante1 =doc.GetItemValue("solicitante_sol")
creador =doc.GetItemValue("creador")
base1 =doc.GetItemValue("base")
server =doc.GetItemValue("server")

'guardar borrador
If (estado_actual(0)="" )Then
    doc.estado = "Por Enviar"
    doc.responsableactual=doc.GetItemValue("resp_actual")
    Call doc.Save(False,False)
    Call email(doc, "Por Enviar", creador(0))
    MsgBox"Por Enviar"
    'Call logestado(doc.resp_actual(0))
End If
```

```

Set db = sesion.GetDatabase(server(0), base1(0))
Set view = db.GetView ("Roles" )
Set docvista = view.GetDocumentByKey ("Director",True)
director= docvista.GetItemValue("nombre_rol")
Set docvista1 = view.GetDocumentByKey ("Decano",True)
decano= docvista1.GetItemValue("nombre_rol")
Set docvista2 = view.GetDocumentByKey ("Administrador",True)
administrador= docvista2.GetItemValue("nombre_rol")
Set docvista3 = view.GetDocumentByKey ("Vice-Rector
Administrativo",True)
viceadministrativo= docvista3.GetItemValue("nombre_rol")
Set docvista4 = view.GetDocumentByKey ("Jefe Adquisiciones",True)
jefeadqui= docvista4.GetItemValue("nombre_rol")
Set docvista5 = view.GetDocumentByKey ("Jefe Activo Fijo",True)
jefeactivo = docvista5.GetItemValue("nombre_rol")
Set docvista6 = view.GetDocumentByKey ("Bodeguero",True)
jefebod = docvista6.GetItemValue("nombre_rol")

'Secretaria avanza
If ((estado_actual(0)="Por Enviar" Or estado_actual(0) ="Por Modificar") And
creador(0)=usuario(0) And boton(0)=1) Then
    doc.estado ="Revision Director"
    doc.responsableactual=director(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Director", director(0))
    Print "<SCRIPT LANGUAGE = 'JavaScript' >
window.history.back();window.alert('Trámite Enviado
a:'+director(0));</SCRIPT>"
End If

```

```
'Director Visto Bueno Avanzar
If (estado_actual(0)="Revision Director" And director(0)=usuario(0) And
boton(0)=2) Then
    doc.estado="Revision Decano"
    doc.responsableactual=decano(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Decano", decano(0))
End If
```

```
'Decano visto bueno avanza
If (estado_actual(0)="Revision Decano" And decano(0)=usuario(0) And
boton(0)=3) Then
    doc.estado = "Revision Administrador"
    doc.responsableactual=administrador(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Administrador", administrador(0))
End If
```

```
'Administrador visto bueno avanza
If (estado_actual(0)="Revision Administrador" And administrador(0)=usuario(0)
And boton(0)=4) Then
    doc.estado = "Revision Vicerrector Administrativo"
    doc.responsableactual=viceadministrativo(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Vicerrector Administrativo", viceadministrativo(0))
End If
```

```
'Vicerrector Administrativo visto bueno avanza
If (estado_actual(0)="Revision Vicerrector Administrativo" And
viceadministrativo(0)=usuario(0) And boton(0)=5) Then
```

```

doc.estado = "Revision Jefe Adquisicion"
doc.responsableactual=jefeadqui(0)
Call doc.Save(False,False)
Call email(doc, "Revision Jefe Adquisicion", jefeadqui(0))
End If

```

```

'Jefe Adquisiciones visto bueno avanza
If (estado_actual(0)="Revision Jefe Adquisicion" And jefeadqui(0)=usuario(0)
And boton(0)=6) Then
    doc.estado = "Revision Jefe Activo"
    doc.responsableactual=jefeactivo(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Jefe Activo", jefeactivo(0))
End If

```

### **EnvioMails**

```

Sub EnvioMails(remitente As String,destinatario As String, asunto As Variant,
mensaje As Variant)
    Set sesion= New NotesSession
    Set db= sesion.CurrentDatabase
    Set DocMens = db.CreateDocument()
    If(DocMens Is Nothing) Then Exit Sub
    DocMens.form="Memo"
    DocMens.subject=asunto
    DocMens.sendto= destinatario
    DocMens.Principal= remitente
    DocMens.Body=mensaje
    Call DocMens.Send(False)
End Sub

```

Los **outlines** permiten crear acciones similares a un menú de opciones. Para este trabajo se utilizaron 2 outlines.

Se los crea en la misma opción Shared Code pero haciendo clic en la palabra Outline y en el botón **New Outline**

Creación de un outline

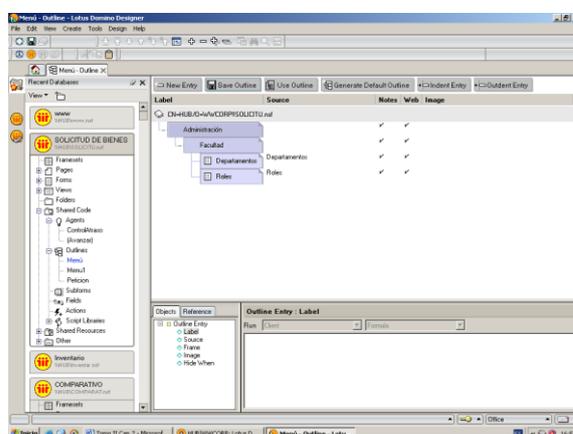


Figura 2.70 Fuente: Lotus Domino Designer 6.5

Estas opciones de los outlines creados son visualizados a los usuarios dependiendo del rol que tenga cada uno.

Los **script libraries** son librerías creadas por los usuarios que contienen funciones que son invocados por medio de agentes o de acciones.

Esta base contiene la librería **Correo:**

**Opciones**

Option Public

**Declarations**

Dim estilo As NotesRichTextStyle

Dim mensaje As NotesRichTextItem

Dim doc\_mail As NotesDocument

Dim db\_mail As NotesDatabase ' base de datos

**Email**

Sub email(doc\_solicitud As NotesDocument, evento As String, destinatario As Variant )

Dim sesion As New NotesSession

Dim doc\_mens As NotesDocument

Dim profiledoc As NotesDocument

Dim rlitm\_mensaje As NotesRichTextItem

Dim rstyle As NotesRichTextStyle

Dim viewruta As NotesView

Dim i As Integer

Dim observaciones As String

Dim server\_perfil, base\_perfil As Variant

Dim pathURL , pathURL1 As Variant

Dim ruta\_mail As String

Dim vistaruta As Notesdocument

Dim dbase As NotesDatabase ' base de datos

Set dbase = sesion.CurrentDatabase

```
server_perfil =doc_solicitud.GetItemValue("server")
base_perfil =doc_solicitud.GetItemValue("base")
Set profiledoc = dbase.GetProfileDocument("Perfil")
'Set pathURL = doc.GetFirstItem("URL")
ruta_mail= profiledoc.URL(0)
pathURL ="http://Hub/SOLICITU.nsf"

Msgbox pathURL

Set rtstyle = sesion.CreateRichTextStyle
Set doc_mens = doc_solicitud.ParentDatabase.CreateDocument()

If doc_mens Is Nothing Then Exit Sub
Set ritm_mensaje = doc_mens.CreateRichTextItem("Body")

If ritm_mensaje Is Nothing Then Exit Sub

doc_mens.Form = "Memo"
doc_mens.SendTo = destinatario
doc_mens.Principal = "PETICION DE BIENES EN UNA FACULTAD"
rtstyle.FontSize = 12
rtstyle.NotesColor = COLOR_BLUE

Msgbox evento

Select Case(evento)
Case "Por Enviar"
    rtstyle.FontSize = 14
    rtstyle.NotesColor = COLOR_BLUE
```

doc\_mens.Subject = "Trámite : "+doc\_solicitud.numero\_sol(0)+" le ha sido enviado para su revisión"

Call rtitm\_mensaje.AppendStyle(rtstyle)

rtitm\_mensaje.AppendText("Se necesita que revise la presente Solicitud de Petición." )

Call rtitm\_mensaje.AddNewLine(2, True)

rtstyle.NotesColor = COLOR\_BLACK

Call rtitm\_mensaje.AppendStyle(rtstyle)

rtitm\_mensaje.AppendText("Fecha de Emisión: "+Cstr(doc\_solicitud.fecha\_sol(0)))

Call rtitm\_mensaje.AddNewLine(1, True)

rtitm\_mensaje.AppendText("Numero de Oficio: "+doc\_solicitud.numero\_sol(0))

Call rtitm\_mensaje.AddNewLine(1, True)

rtitm\_mensaje.AppendText("Solicitante: "+doc\_solicitud.solicitante\_sol(0))

Call rtitm\_mensaje.AddNewLine(1, True)

rtitm\_mensaje.AppendText("Departamento:

" +doc\_solicitud.departamento\_sol(0))

Call rtitm\_mensaje.AddNewLine(1, True)

rtitm\_mensaje.AppendText("Facultad: "+doc\_solicitud.facultad\_sol(0))

#### Case "Revision Director"

doc\_mens.Subject = "Trámite : "+doc\_solicitud.numero\_sol(0)+" le ha sido enviado para su revisión"

rtstyle.NotesColor = COLOR\_BLACK

Call rtitm\_mensaje.AppendStyle(rtstyle)

rtitm\_mensaje.AppendText("Se necesita que revise el presente Trámite." )

Call rtitm\_mensaje.AddNewLine(2, True)

rtstyle.NotesColor = COLOR\_BLACK

Call rtitm\_mensaje.AppendStyle(rtstyle)

```

rtitm_mensaje.AppendText("Fecha de Emisión: "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Revision Decano"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise el presente Trámite." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha de Emisión: "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Revision Administrador"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise el presente Trámite." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha      de      Emisión:      "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero      de      Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Revision Vicerrector Administrativo"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"

rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise el presente Trámite." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha      de      Emisión:      "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero      de      Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Revision Jefe Adquisicion"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"

rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise el presente Trámite." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)

```

```

rtitm_mensaje.AppendText("Fecha de Emisión: "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Revision Jefe Activo"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise el presente Trámite." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha de Emisión: "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Por Modificar"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que modifique el siguiente Trámite."
)
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha      de      Emisión:      "+Cstr(
doc_solicitud.fecha_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero      de      Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

```

#### Case "Solicitud Rechazada"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numero_sol(0)+" le ha sido
enviado para su revisión"

rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("La Solicitud de peticiones ha sido Rechazada." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Fecha      de      Emisión:      "+Cstr(
doc_solicitud.fecha_sol(0)))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero      de      Oficio:
"+doc_solicitud.numero_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Solicitante: "+doc_solicitud.solicitante_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Departamento:
"+doc_solicitud.departamento_sol(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad: "+doc_solicitud.facultad_sol(0))

Case Else
    rtitm_mensaje.AppendText("No hay documento por Revisar" )
End Select

rtstyle.FontSize = 13
rtstyle.NotesColor = COLOR_BLUE
Call rtitm_mensaje.AppendStyle(rtstyle)
Call rtitm_mensaje.AddNewLine(2, True)

```

```
Call rtitm_mensaje.AppendText("Para la Revisión del Trámite a procesar ")
Call rtitm_mensaje.AddNewLine(1, True)
Call rtitm_mensaje.AppendText("Use su navegador de Internet e ingrese a la
direccion "+pathURL+"/0/"+doc_solicitud.UniversallID+"?OpenDocument" )
Call rtitm_mensaje.AddNewLine(1, True)
Call rtitm_mensaje.AppendText("Allí encontrará el documento para su revisión.")
Call rtitm_mensaje.AddNewLine(1, True)
Call doc_mens.Send(True)
```

End Sub

## 2.2.2. Base COMPARAT.nsf

### 2.2.2.1. Creación de forms

Para esta base hemos creado 1 formularios básicos y 1 formularios basados en una plantilla existente.

- Solicitudes Comparativo
- \$\$View Template for Mis Solicitudes

Cada uno de ellos contiene campos, vistas, secciones, y acciones que permiten controlar el proceso del flujo del workflow.

El valor de los campos son modificados en el **default value** en la ficha de objetos del panel de programación, o accediendo a las propiedades del campo haciendo clic derecho sobre el campo y escogiendo la opción **Field Properties**.

En la acción de Imprimir se utilizó el siguiente código.

```
<SCRIPT LANGUAGE='JavaScript'>
<!--
function imprimir() {
version = parseInt(navigator.appVersion);
if (version >= 4)
window.print();
}
// --></SCRIPT>
```

A continuación describimos el código para cada formulario

- **Formulario Solicitudes\_Comparativo**

### Campos usados

Campo	Tipo de Campo	Valor del Campo
Num_oficio	Text	Uso de lenguaje de fórmula <b>@UriQueryString("numoficio")</b>
area_solicitante	Text	Uso de lenguaje de fórmula <b>@UriQueryString("area_solicitante")</b>
Empresa1	Text	Ingreso de información
Empresa2	Text	Ingreso de información
Empresa3	Text	Ingreso de información
Cantidad#	Number	Ingreso de información
Destino#	Text	Ingreso de información
Valor#	Number	Ingreso de información
Total#	Number	<b>Campo calculado mediante acción</b>
Sub#	Number	<b>Campo calculado mediante acción</b>
IVA#	Number	<b>Campo calculado mediante acción</b>
Gana	Number	<b>Campo calculado mediante acción</b>
body	Rich text	<b>Para anexas archivos al documento</b>

Tabla 2.26

Fuente: Grupo de Proyecto

### Acciones

Accion	Código. Uso de lenguaje de fórmula
Guardar	<b>@Command([FileSave])</b>
Comparar	<b>REM{@If(1;sub1 &amp; sub2 &amp; sub3:= 0;sub1:=sub1 &amp; sub2:=sub2 &amp; sub3:=sub3);}; sub1 := 0; sub2 :=0; sub3 :=0; num_sol:=@Text(numoficio);</b>
Regresar	<b>@URLOpen("http://hub/SOLICITU.nsf/Solicitudes/" +num_sol+"?OpenDocument")</b>
Imprimir	<b>JavaScript Imprimir()</b>

Tabla 2.27

Fuente: Grupo de Proyecto

- **Formulario \$View Template for Mis Solicitudes**

### Vistas Embebidas

Vista	Código
Comparativo	<b>Embedded Selection</b>

Tabla 2.28

Fuente: Grupo de Proyecto

### Formulario cuadro comparativo

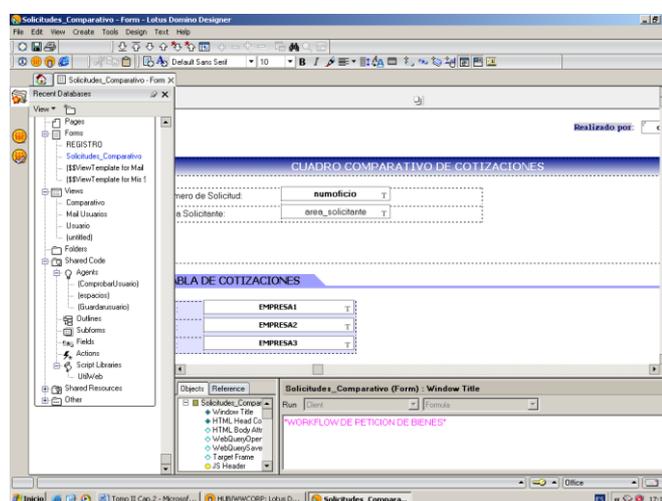


Figura 2.71 Fuente: Lotus Domino Designer 6.5

### 2.2.2.2. Creación de Vistas

Para nuestro workflow hemos creado 2 vistas. Cada vista esta relacionada con un formulario.

Vista de Solicitudes

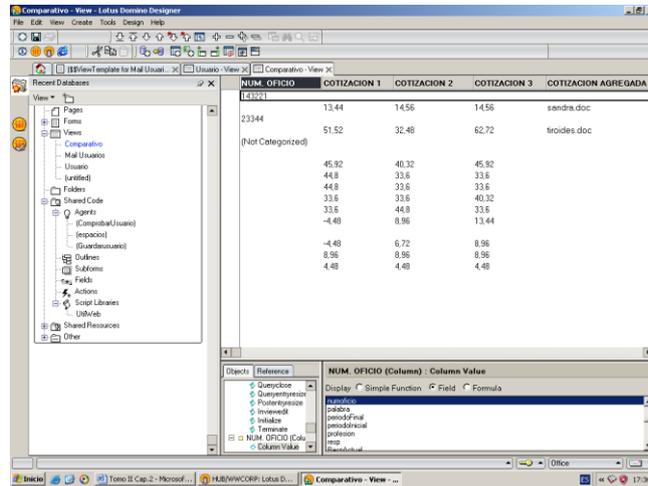


Figura 2.72 Fuente: Lotus Domino Designer 6.5

Vista Comparativo

Vista	View Selection	Campos-Columnas
Comparativo	SELECT form ="Solicitudes_Comparativo"	Cada columna esta relacionada con un campo

Tabla 2.29 Fuente: Grupo de Proyecto

### 2.2.3. Base Inventar.nsf

Esta base de datos fue diseñada para la codificación y asignación de los bienes adquiridos de la Facultad, a continuación se describe como fue elaborada la base de datos.

#### 2.2.3.1. Creación de forms

Para esta base de datos hemos creado 1 formularios básicos y 1 formularios basados en una plantilla existente.

- **Formulario Inventario**

Creación del formulario inventario

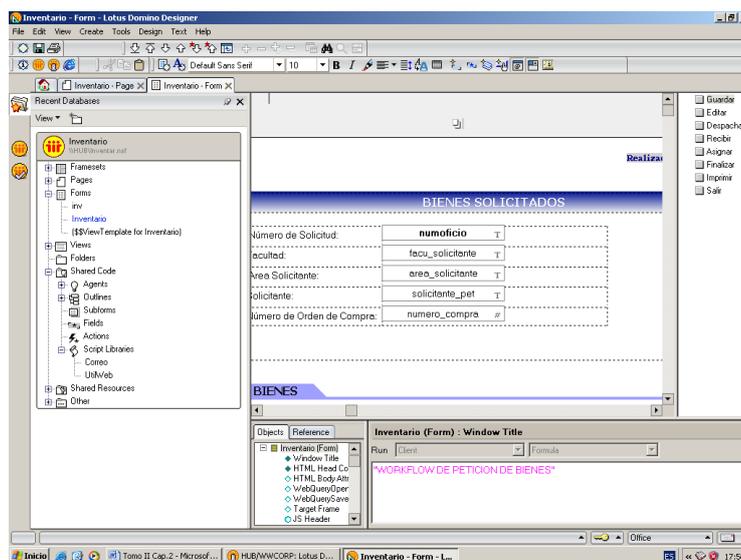


Figura 2.73 Fuente: Lotus Domino Designer

Este formulario contiene campos, vistas, secciones, y acciones que permiten controlar el proceso del flujo del workflow. El valor de los campos son modificados en el **default value** en la ficha de objetos del

panel de programación, o accediendo a las propiedades del campo haciendo clic derecho sobre el campo y escogiendo la opción **Field Properties**.

En la acción Imprimir se utilizó el siguiente código Script.

```
<SCRIPT LANGUAGE='JavaScript'>
<!--
function imprimir() {
version = parseInt(navigator.appVersion);
if (version >= 4)
window.print();
}
// --></SCRIPT>
```

A continuación describimos el código para del formulario

- **Formulario Inventario**

### Campos usados

Campo	Tipo de Campo	Valor del Campo
<b>Numoficio</b>	Text	Uso de lenguaje de fórmula <b>@QueryString("numoficio")</b>
<b>Facu_solicitante</b>	text	Uso de lenguaje de fórmula <b>@QueryString("facu_solicitante")</b>
<b>Area_solicitante</b>	Text	Uso de lenguaje de fórmula <b>@QueryString("area_solicitante")</b>
<b>Solicitante_pet</b>	Text	Uso de lenguaje de fórmula <b>@QueryString("solicitante_pet")</b>
<b>Numero_compra</b>	Number	Ingreso de información
<b>Codigo#</b>	Number	Ingreso de información
<b>Tipo_bien#</b>	Dialog list	Lista de Opciones
<b>Descripción</b>	Text	Ingreso de información
<b>Ubicación</b>	Text	Ingreso de información
<b>responsable</b>	Text	Ingreso de información

Tabla 2.30

Fuente: Grupo de Proyecto

## Acciones

Accion	Código
<b>Guardar</b>	Uso de lenguaje de fórmula <b>FIELD boton:=0;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Flujo)")</b>
<b>Editar</b>	Uso de lenguaje de fórmula <b>@Command([EditDocument])</b>
<b>Despachar</b>	Uso de lenguaje de fórmula <b>FIELD boton:=1;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Flujo)")</b>
<b>Recibir</b>	Uso de lenguaje de fórmula <b>FIELD boton:=2;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Flujo)")</b>
<b>Asignar</b>	Uso de lenguaje de fórmula <b>FIELD boton:=3;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Flujo)")</b>
<b>Finalizar</b>	Uso de lenguaje de fórmula <b>FIELD boton:=4;</b> <b>@Command([FileSave]);</b> <b>@Command([ToolsRunMacro]; "(Flujo)")</b>
<b>Imprimir</b>	<b>Imprimir()</b>
<b>Salir</b>	Uso de lenguaje de fórmula <b>@SetTargetFrame("central");</b> <b>@URLOpen("http://Hub/SOLICITU.nsf/Salida?OpenPage")</b>

Tabla 2.31

Fuente: Grupo de Proyecto

- **Formulario \$\$View Template for Inventario**

Vista	Código
Inventario	<b>Embedded Selection</b>
	<b>Show Single Category: usuario</b>
Campos	Ingreso de Información

Tabla 2.32

Fuente: Grupo de Proyecto

### 2.2.3.2. Creación de view

Para nuestro workflow hemos creado 2 vistas. Está relacionada con un formulario lo cual se codifica en el View Selection de la sección objetos del panel de programación.



### 2.2.3.3. Creación de shared code

Se realizaron 2 Agentes

1. **ArchivoExcel**, permite generar un reporte en Excel de los bienes codificados

#### Options

Option Public

Option Declare

Use "UtilWeb"

#### Initialize

Sub Initialize

Dim estado\_actual As Variant

Dim logitem As Variant

Dim responsable, responsableactual As Variant

Dim director, decano, viceadministrativo, jefeadqui As Variant

Dim jefeactivo, administrador, jefebod As Variant

Dim usuario As Variant

Dim creador, docid As Variant

Dim nombre As String

Dim view As NotesView

Dim boton As Variant

Dim server, base1 As Variant

Set sesion = New NotesSession

Set doc = sesion.DocumentContext

Dim docvista, docvista1, docvista2, docvista3, docvista4, docvista5, docvista6  
As Notesdocument

Dim item As NotesItem

```
Set db = sesion.currentdatabase

docid = doc.GetItemValue("id")
estado_actual=doc.GetItemValue("estado")
boton=doc.GetItemValue("boton")
usuario = doc.GetItemValue("resp_actual")
'solicitante1 =doc.GetItemValue("solicitante_sol")
creador =doc.GetItemValue("creador")
base1 =doc.GetItemValue("base")
server =doc.GetItemValue("server")

'guardar borrador
If (estado_actual(0)="" )Then
    doc.estado = "Por Enviar"
    doc.responsableactual=doc.GetItemValue("resp_actual")
    Call doc.Save(False,False)
    Call email(doc, "Por Enviar", creador(0))
    MsgBox"Por Enviar"
    'Call logestado(doc.resp_actual(0))
End If

Set db = sesion.GetDatabase(server(0), base1(0))
Set view = db.GetView ("Roles" )
Set docvista = view.GetDocumentByKey ("Director",True)
director= docvista.GetItemValue("nombre_rol")
Set docvista1 = view.GetDocumentByKey ("Decano",True)
decano= docvista1.GetItemValue("nombre_rol")
Set docvista2 = view.GetDocumentByKey ("Administrador",True)
administrador= docvista2.GetItemValue("nombre_rol")
```

```

Set docvista3 = view.GetDocumentByKey ("Vice-Rector Administrativo",True)
    viceadministrativo= docvista3.GetItemValue("nombre_rol")
Set docvista4 = view.GetDocumentByKey ("Jefe Adquisiciones",True)
    jefeadqui= docvista4.GetItemValue("nombre_rol")
Set docvista5 = view.GetDocumentByKey ("Jefe Activo Fijo",True)
    jefeactivo = docvista5.GetItemValue("nombre_rol")
Set docvista6 = view.GetDocumentByKey ("Bodeguero",True)
    jefebod = docvista6.GetItemValue("nombre_rol")

```

'Secretaria avanza

```

If ((estado_actual(0)="Por Enviar" Or estado_actual(0) ="Por Modificar") And
creador(0)=usuario(0) And boton(0)=1) Then
    doc.estado ="Revision Director"
    doc.responsableactual=director(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Director", director(0))
Print    "<SCRIPT    LANGUAGE    =    'JavaScript'    >
window.history.back();window.alert('Trámite    Enviado
a:'+director(0));</SCRIPT>"
End If

```

'Director Visto Bueno Avanzar

```

If (estado_actual(0)="Revision Director" And director(0)=usuario(0) And
boton(0)=2) Then
    doc.estado="Revision Decano"
    doc.responsableactual=decano(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Decano", decano(0))
End If

```

```

'Decano visto bueno avanza
If (estado_actual(0)="Revision Decano" And decano(0)=usuario(0) And
boton(0)=3) Then
    doc.estado = "Revision Administrador"
    doc.responsableactual=administrador(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Administrador", administrador(0))
End If

'Administrador visto bueno avanza
If (estado_actual(0)="Revision Administrador" And administrador(0)=usuario(0)
And boton(0)=4) Then
    doc.estado = "Revision Vicerrector Administrativo"
    doc.responsableactual=viceadministrativo(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Vicerrector Administrativo", viceadministrativo(0))
End If

'Vicerrector Administrativo visto bueno avanza
If (estado_actual(0)="Revision Vicerrector Administrativo" And
viceadministrativo(0)=usuario(0) And boton(0)=5) Then
    doc.estado = "Revision Jefe Adquisicion"
    doc.responsableactual=jefeadqui(0)
    Call doc.Save(False,False)
    Call email(doc, "Revision Jefe Adquisicion", jefeadqui(0))
End If

'Jefe Adquisiciones visto bueno avanza
If (estado_actual(0)="Revision Jefe Adquisicion" And jefeadqui(0)=usuario(0)
And boton(0)=6) Then
    doc.estado = "Revision Jefe Activo"
    doc.responsableactual=jefeactivo(0)
    Call doc.Save(False,False)

```

Call email(doc, "Revision Jefe Activo", jefeactivo(0))

End If

## 2. Flujo, crea el flujo para la base de datos inventario

### Options

Option Public

Option Declare

Use "Correo"

### Declarations

Dim sesion As NotesSession

Dim doc As NotesDocument

Dim db As NotesDatabase

Dim DocMens As NotesDocument

Dim logitem As Variant

### Initialize

Sub Initialize

Dim estado\_actual As Variant

Dim logitem As Variant

Dim responsable, responsableactual As Variant

Dim director, decano, viceadministrativo, jefeadqui As Variant

Dim jefeactivo, administrador, jefebod As Variant

Dim usuario As Variant

Dim creador, docid As Variant

Dim nombre As String

Dim view As NotesView

Dim boton As Variant

Dim server, base1 As Variant

```
Set sesion = New NotesSession
Set doc = sesion.DocumentContext

Dim docvista, docvista1, docvista2, docvista3, docvista4, docvista5,
docvista6 As Notesdocument

Dim item As NotesItem

Set db = sesion.currentdatabase

docid = doc.GetItemValue("id")
estado_actual=doc.GetItemValue("estado")
boton=doc.GetItemValue("boton")
usuario = doc.GetItemValue("resp_actual")
'solicitante1 =doc.GetItemValue("solicitante_sol")
creador =doc.GetItemValue("creador")
base1 =doc.GetItemValue("base")
server =doc.GetItemValue("server")

'guardar borrador

If (estado_actual(0)="") Then
    doc.estado="Por Entregar"
    doc.responsableactual=doc.GetItemValue("resp_actual")
    Call doc.Save(False,False)
    'Call logestado(doc.resp_actual(0))
End If

Set db = sesion.GetDatabase("HUB", "SOLICITU.nsf")
Set view = db.GetView ("Roles" )
Set docvista2 = view.GetDocumentByKey ("Administrador",True)
```

```

administrador= docvista2.GetItemValue("nombre_rol")
Set docvista2 = view.GetDocumentByKey ("Director",True)
director= docvista2.GetItemValue("nombre_rol")
Set docvista5 = view.GetDocumentByKey ("Jefe Activo Fijo",True)
jefeactivo = docvista5.GetItemValue("nombre_rol")
Set docvista6 = view.GetDocumentByKey ("Bodeguero",True)
jefebod = docvista6.GetItemValue("nombre_rol")

'Jefe de Activo avanza
If (estado_actual(0)="Por Entregar" And creador(0)=usuario(0) And
boton(0)=1) Then
    doc.estado="Por Recibir"
    doc.responsableactual=administrador(0)
    Call doc.Save(False,False)
    Call email(doc, "Por Recibir", administrador(0))

End If

If (estado_actual(0)="Por Recibir" And administrador(0)=usuario(0) And
boton(0)=2) Then
    doc.estado="Recibido Administrador"
    doc.responsableactual=jefebod(0)
    Call doc.Save(False,False)
    Call email(doc, "Recibido Administrador", jefebod(0))

End If

If (estado_actual(0)="Recibido Administrador" And jefebod(0)=usuario(0)
And boton(0)=3) Then
    doc.estado="Recibido Bodeguero"
    doc.responsableactual=director(0)
    Call doc.Save(False,False)

```

```
        Call email(doc, "Recibido Bodeguero", director(0))
    End If

    If (estado_actual(0)="Recibido Bodeguero" And director(0)=usuario(0) And
    boton(0)=4) Then
        doc.estado = "Tramite Finalizado"
        Call doc.Save(False,False)
        Call email(doc, "Trámite Finalizado", administrador(0))

    End If

End Sub
```

### **EnvioMail**

```
Sub EnvioMails(remitente As String,destinatario As String, asunto As Variant,
mensaje As Variant)

    Set sesion= New NotesSession
    Set db= sesion.CurrentDatabase
    Set DocMens = db.CreateDocument()

    If(DocMens Is Nothing) Then Exit Sub
    DocMens.form="Memo"
    DocMens.subject=asunto
    DocMens.sendto= destinatario
    DocMens.Principal= remitente
    DocMens.Body=mensaje
    Call DocMens.Send(False)

End Sub
```

Los **script libraries** son librerías creadas por los usuarios que contienen funciones que son invocados por medio de agentes o de acciones.

Esta base contiene la librería **Correo:**

#### Opciones

Option Public

#### Declarations

Dim estilo As NotesRichTextStyle

Dim mensaje As NotesRichTextItem

Dim doc\_mail As NotesDocument

Dim db\_mail As NotesDatabase ' base de datos

#### Email

Sub email(doc\_solicitud As NotesDocument, evento As String, destinatario As Variant )

Dim sesion As New NotesSession

Dim doc\_mens As NotesDocument

Dim profiledoc As NotesDocument

Dim rtitm\_mensaje As NotesRichTextItem

Dim rtstyle As NotesRichTextStyle

Dim viewruta As NotesView

Dim i As Integer

Dim observaciones As String

Dim server\_perfil, base\_perfil As Variant

Dim pathURL , pathURL1 As Variant

Dim ruta\_mail As String

```
Dim vistaruta As Notesdocument
```

```
Dim dbase As NotesDatabase ' base de datos
```

```
Set dbase = sesion.CurrentDatabase
```

```
server_perfil =doc_solicitud.GetItemValue("server")
```

```
base_perfil =doc_solicitud.GetItemValue("server")
```

```
Set profiledoc = dbase.GetProfileDocument("Perfil")
```

```
'Set pathURL = doc.GetFirstItem("URL")
```

```
ruta_mail= profiledoc.URL(0)
```

```
pathURL ="http://Hub/Inventar.nsf"
```

```
Set rtstyle = sesion.CreateRichTextStyle
```

```
Set doc_mens = doc_solicitud.ParentDatabase.CreateDocument()
```

```
If doc_mens Is Nothing Then Exit Sub
```

```
Set rtitm_mensaje = doc_mens.CreateRichTextItem("Body")
```

```
If rtitm_mensaje Is Nothing Then Exit Sub
```

```
doc_mens.Form = "Memo"
```

```
doc_mens.SendTo = destinatario
```

```
doc_mens.Principal = "PETICION DE BIENES EN UNA FACULTAD"
```

```
rtstyle.FontSize = 14
```

```
rtstyle.NotesColor = COLOR_DARK_BLUE
```

```
Msgbox evento
```

```
Select Case(evento)
```

```
Case "Por Entregar"
```

```
doc_mens.Subject = "Trámite : "+doc_solicitud.numoficio(0)+" le ha sido
enviado para su revisión"
```

```
rtstyle.FontSize = 12
```

```
rtstyle.NotesColor = COLOR_BLACK
```

```
Call rtitm_mensaje.AppendStyle(rtstyle)
```

```
rtitm_mensaje.AppendText("Se necesita que revise la presente Solicitud de
Petición." )
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtstyle.NotesColor = COLOR_BLACK
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtitm_mensaje.AppendText("Numero de Oficio: "+doc_solicitud.numoficio(0))
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtitm_mensaje.AppendText("Facultad                               Solicitante:
"+doc_solicitud.facu_solicitante(0))
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtitm_mensaje.AppendText("Area                               Solicitante:
"+doc_solicitud.area_solicitante(0))
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtitm_mensaje.AppendText("Número                               de           Compra:
"+doc_solicitud.numero_compra(0))
```

#### Case "Revision Bodeguero"

```
doc_mens.Subject = "Trámite : "+doc_solicitud.numoficio(0)+" le ha sido
enviado para su revisión"
```

```
rtstyle.NotesColor = COLOR_BLACK
```

```
Call rtitm_mensaje.AppendStyle(rtstyle)
```

```
rtitm_mensaje.AppendText("Se necesita que revise la presente Solicitud de
Petición." )
```

```
Call rtitm_mensaje.AddNewLine(2, True)
```

```
rtstyle.NotesColor = COLOR_BLACK
```

```

Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio: "+doc_solicitud.numoficio(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad                               Solicitante:
"+doc_solicitud.facu_solicitante(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Area                               Solicitante:
"+doc_solicitud.area_solicitante(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Número                               de                               Compra:
"+doc_solicitud.numero_compra(0))

```

Case "Revision Administrador"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numoficio(0)+" le ha sido
enviado para su revisión"
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AppendStyle(rtstyle)
rtitm_mensaje.AppendText("Se necesita que revise la presente Solicitud de
Petición." )
Call rtitm_mensaje.AddNewLine(2, True)
rtstyle.NotesColor = COLOR_BLACK
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Numero de Oficio: "+doc_solicitud.numoficio(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Facultad                               Solicitante:
"+doc_solicitud.facu_solicitante(0))
Call rtitm_mensaje.AddNewLine(2, True)
rtitm_mensaje.AppendText("Area                               Solicitante:
"+doc_solicitud.area_solicitante(0))
Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Número           de           Compra:
"+doc_solicitud.numero_compra(0))

```

Case "Tramite Finalizado"

```

doc_mens.Subject = "Trámite : "+doc_solicitud.numoficio(0)+" le ha sido
enviado para su revisión"

```

```

rtstyle.NotesColor = COLOR_RED

```

```

Call rtitm_mensaje.AppendStyle(rtstyle)

```

```

rtitm_mensaje.AppendText("Se necesita que revise la presente Solicitud de
Petición." )

```

```

Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtstyle.NotesColor = COLOR_BLACK

```

```

Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Numero de Oficio: "+doc_solicitud.numoficio(0))

```

```

Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Facultad           Solicitante:

```

```

"+doc_solicitud.facu_solicitante(0))

```

```

Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Area           Solicitante:

```

```

"+doc_solicitud.area_solicitante(0))

```

```

Call rtitm_mensaje.AddNewLine(2, True)

```

```

rtitm_mensaje.AppendText("Número           de           Compra:

```

```

"+doc_solicitud.numero_compra(0))

```

Case Else

End Select

```

rtstyle.FontSize = 12

```

```

rtstyle.NotesColor = COLOR_BLUE

```

```

Call rtitm_mensaje.AppendStyle(rtstyle)
Call rtitm_mensaje.AddNewLine(2, True)
Call rtitm_mensaje.AppendText("Use su navegador de Internet e ingrese a la
direccion "+pathURL+"/0/"+doc_solicitud.UniversallID+"?OpenDocument" )
Call rtitm_mensaje.AddNewLine(1, True)
Call rtitm_mensaje.AppendText("Allí encontrará el documento para su
revisión."+pathURL1)
Call rtitm_mensaje.AddNewLine(1, True)

Call doc_mens.Send(True)

```

End Sub

### **Correoestados**

Sub correo\_estados(subject As String, destinatario As Variant, remitente As Variant )

```

Dim session As New NotesSession
Dim profiledoc As NotesDocument
Dim pathURL As Variant

Set session = New NotesSession
Set db_mail = session.CurrentDatabase
'Set doc_mail = db.CreateDocument()
Set doc_mail=session.DocumentContext
Set mensaje=doc_mail.CreateRichTextItem("Body")
Set estilo= session.CreateRichTextStyle ' se llena el richstyle, es un constructor
estilo.FontSize = 12
estilo.NotesColor = COLOR_RED

'Set profiledoc = db.GetProfileDocument("Perfil")

```

```

pathURL = hub/Peticiones.nsf
'profiledoc.URLCompleto(0)

doc_mail.Form="Memo"
doc_mail.Subject = subject
doc_mail.sendTo = destinatario ' sendto reemplaza al segundo parámetro del método
[recipient]
doc_mail.Principal= remitente 'o DocMens.RcptTo ="Pepe Perez" // es lo mismo
'call mensaje.Body ="Solicitud enviada para su Revisión"

Call mensaje.AppendStyle(estilo) ' se asocia el notesrichtextitem al estilo creado
Call mensaje.AppendText(subject)
Call mensaje.AddNewline(1,True)
estilo.FontSize = 12
estilo.NotesColor = COLOR_BLUE
Call mensaje.AddNewLine (2,True)
Call mensaje.AppendText("El estado actual del proceso es: " +subject)
Call mensaje.AppendText(Chr(13) + "El vinculo para acceder a el -> ")
Call mensaje.AppendDocLink(doc_mail,db_mail.Title)
Call doc_mail.Send(False)

End Sub

```

### 2.3. Creación de shared resource para todas las bases

Se utilizaron imágenes gif para personalizar la presentación de la aplicación y se crearon 2 hojas de estilo las cuales permiten tener un formato estándar de los elementos de diseño utilizados en la aplicación

### 2.3.1. Peticiones.css

```
body{  
    font-family: "Bookman Old Style";  
    margin: 5px;  
    padding: 5px;  
    font-style: normal;  
    font-size: 11px;  
    background: #E8E8E8;  
}
```

```
Table {  
    background : transparent;  
    border: 1px Black;  
    border-spacing: 0px;  
    border-collapse: collapse;  
    border-color: Black;  
    border-bottom-width: 2px;  
    border-bottom: 1px solid Black;  
}
```

```
Th{  
    font-family: "MS Sans Serif";  
    font-size: 12px;  
    color: White;  
    background: Navy;  
    height: 18px;  
    font: normal normal;  
    border: 1px solid Black;  
}
```

```
Td {
```

```
        font-family: "MS Sans Serif";
        font-size: 12px;
        border: 0px;
        vertical-align: middle;
        padding: 1px;
        background-color: #ACC ACC;
    }
    tr{
        vertical-align: middle;
    }
    input{
        font-family: "Bookman Old Style";
        color: Black;
        font-size: 12px;
        border: 1px solid;
        height: 18px;
        width: auto;
        background: White;
    }
    Hr{
        color: Black;
        height: 1px;
        margin-top: -5px;
    }
    .select{
        font-family: Verdana, Arial,sans-serif;
        font-size: 11px;
        border: 1px solid rgb(0,130,191);
        height: 18px;
        color: #003500;
```

```
        background: transparent;
    }
    .pageCelda{
        position: absolute;
        top: 115px;
        left: 10%;
        right: 2%;
        width: 78.6%;
        background-color: White;
        border: 2px ridge Black;
    }
    .marco{
        position: absolute;
        top: 130px;
        left: 10%;
        right: 10%;
        width: 80%;
        background-color: White;
        border: 1px solid Black;
    }

    .Table{
        width: 600px;
        background: Navy;
        border: 1px solid Black;
        margin-top: 4%;
        margin-left: 10%;
        margin-right: 5%;
    }
    .TableDatos{
```

```
width: 600px;
    background : transparent;
    /*border: 2px solid rgb(0,130,191);*/
    margin-top:4%;
    margin-left: 10%;
    margin-right: 5%;

}

.datosTH{
    font-family : "Bookman Old Style";
    font-size : 15px;
    color: White;
    font: bolder;
    height: 23px;
    background: Navy;
    }

.datosTD{
    font-family: "Bookman Old Style";
    font-size: 11px;
    font-weight: 600;
    color: Black;
    height: 20px;
    background: White;
    border: 1px solid #0A0A0A;
}

.tituloTable{
width: 600px;
```

```
height: 30px;
margin-top: 2%;
margin-left: 10%;
margin-right: 5%;
border: 1px solid Navy;
font-family: Tahoma,Verdana, Arial,sans-serif;
font-size: 16px;
font-weight: bold;
color: #0A0A0A;
font-size-adjust: 1;
}

.TituloTD{
    font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: bold;
color: #FDFDFD;
    background: Navy;
}

}

a:link, a:visited {
    text-decoration: none;
display: block;
border-top: 1px dashed Black;
border-bottom: 1px dashed Black;
padding: 1px 4px 3px 3px;
color: black;
border: 1px solid Black;
}
}
```

```
a:hover{
    border-top: 1px solid Black;
    border-bottom: 1px solid Black;
    background-color: Navy;
    background-image: none; /*font-weight: bold;*/
    text-decoration: none;
    color: white;
    border: 1px solid White;
}

/***** header tag styles *****/
img{
    vertical-align: middle;
}
h1{
    font: bold 120% Arial,sans-serif;
    color: #334d55;
    margin: 0px;
    padding: 0px;
}
h2{
    font: bold 114% Arial,sans-serif;
    color: #006699;
    margin: 0px;
    padding: 0px;
}
```

```
h3{
font: bold 100% Arial,sans-serif;
color: #334d55;
margin: 0px;
padding: 0px;
color: White;
}
```

```
h4{
font: 100% Arial,sans-serif;
color: #333333;
margin: 0px;
padding: 0px;
}
```

```
h5{
font: 100% Arial,sans-serif;
color: #334d55;
margin: 0px;
padding: 0px;
}
```

```
/****** list tag styles *****/
```

```
ul{
list-style-type: square;
}
```

```
form {
```

```

        margin: 0;
        padding: 0;
    }

    label{
        font: bold 1em Arial,sans-serif;
        color: Black;
        border: 1px dashed #696969;
    }

```

```

.seccion{
    width: 600px;
    background : transparent;
    /*border:#104a7b 1px solid;*/
    margin-top:2%;
    margin-left: 10%;
    margin-right: 5%;
    font-family: Verdana, Arial,sans-serif;
    font-size: 12px;
    /*color: #96965E;*/
    border: 0px;
    font-size-adjust: 1;
}

```

### 2.3.2 View.css

/\* Hoja de estilo para las vistas en la Web \*/

```

body{
    font-family: "Bookman Old Style"; /*line-height: 1.166; */
    margin: 5px;
    padding: 5px;
}

```

```
        font-style: normal;
        font-size: 11px;
        background: #DEDEDE;
    }
    Table {
        background: #EBF1F7;
        border: 1px double Black; /*border-spacing: 0px;*/
        border-collapse: collapse;
        color: Black;
    }
    TH{
        font-size: 14px;
        border-top: 1px #1C3248;
        border-left: thin groove #1C3248;
        border-right: thin groove #1C3248;
        border-bottom: thin none #1C3248;
        background: #1C3248;
        height: 25px;
        padding-left: 12px;
        color: #FEFEFE;
    }

    Tr{
        vertical-align: middle;
    }
    TD {
        font-family: verdana, Arial, sans-serif;
        font-size: 11px;
        height: 18px; /*background: #EEF6FA;*/
        background-color: transparent;
```

```
padding: 2px;
border-bottom: 1px solid #1C3248;
vertical-align: middle;
color: Black;
}
Hr{
    color: Black;
    height: 1px;
    margin-top: -5px;
}
a:link, a:visited {
text-decoration: none;
display: block;
padding: 2px 4px 3px 3px;
/*color: #21536A;*/
color: Black;
font: normal;
}

a:hover{
background-color: Aqua;
background-image: none; /*font-weight: bold;*/
text-decoration: none;
color: #050505;
}

img{
vertical-align: middle;
}
```

## **2.4. Accesos y Permisos de los Usuarios**

Lotus Domino brinda seguridades a las diferentes aplicaciones creadas o que esten por crearse.

En el workflow de petición de bienes contamos con las siguientes seguridades.

### **2.4.1. ACL**

Lista de control de accesos (Access Control List ), una vez que la red y el servidor admiten al usuario, este se encontrará con restricciones o controles de acceso a nivel de base de datos.

Por lo general esta tarea de configuración y mantenimiento de este nivel lo realizan los administradores de las bases de datos a través del ACL.

Para acceder al ACL, ingrese al Menú File, opción Database, opción Access Control... o haga clic derecho sobre la base de datos a la cual quiera modificar el ACL, y aparecerá el siguiente cuadro de diálogo.

Ingrese a los usuarios a los que va a darles los permisos.

## Permisos del ACL

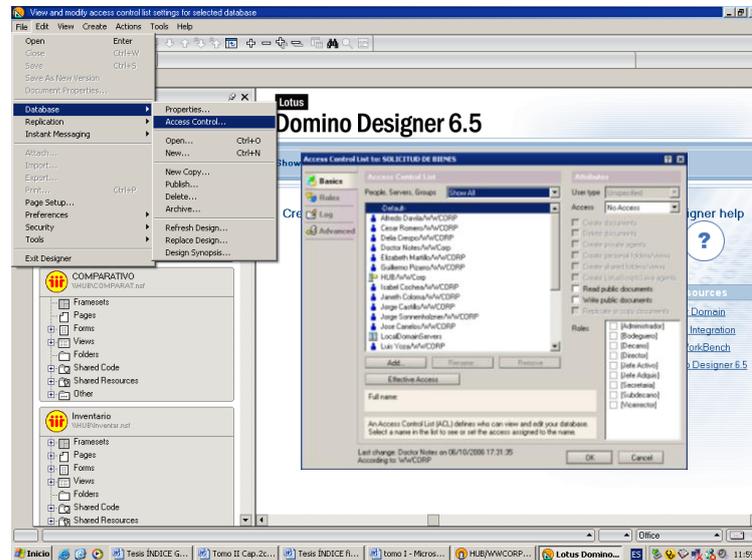


Figura 2.75 Fuente Lotus Domino Designer 6.5

### 2.4.2. Roles

Una vez que los usuarios acceden a la base de datos, las medidas de seguridad pueden limitar el acceso a los datos y los elementos de diseño como por ejemplo, documentos, formularios y vistas.

Los roles proporcionan un método flexible para restringir el acceso a un determinado grupo de usuarios, se pueden utilizar en las fórmulas, no es necesario mantenerlos ni incluirlos en el Directorio de Domino de la organización. Además evitan tener que modificar la aplicación cada vez que se producen bajas o nuevas incorporaciones en la organización.

Para ello acceda al ACL y haga clic en el tab Roles y agregue los roles que debe tener cada usuario.

#### Roles asignados a los Usuarios

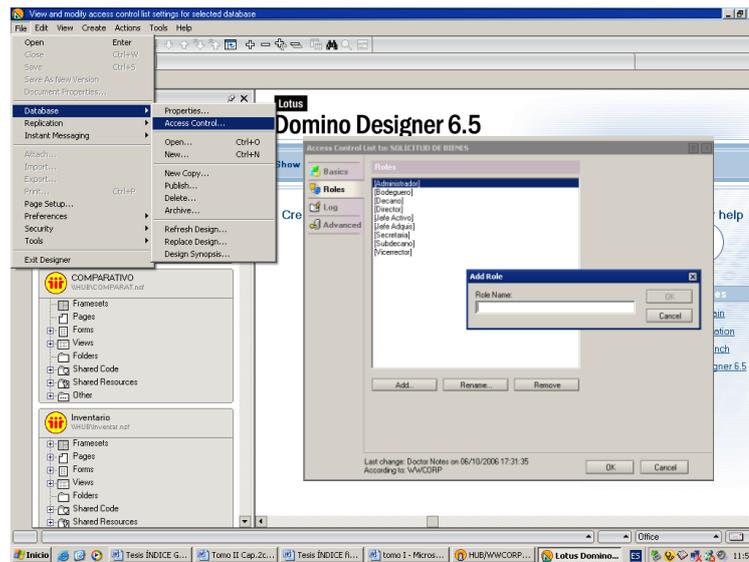


Figura 2.76

Fuente: Lotus Domino Designer 6.5

#### Usuarios Creados con Roles Asignados

Usuario	ACL	Roles
Nilda Ronquillo	Editor	SECRETARIA
Alfredo Davila	Editor	DIRECTOR
Noemí Cruz	Editor	SECRETARIA
Isabel Cochea	Editor	SECRETARIA
Wilson Maitta	Editor	DECANO
Mercy Sanchez	Editor	SECRE_ADMIN
Jose Canelos	Manager	ADMINISTRADOR
Cesar Romero	Editor	VICERRECTOR
Vicente Paz	Editor	JEFE_ADQUIS
Luis Yoza	Editor	JEFE_ACTIVO
Miguel Tenelanda	Editor	BODEGUERO

Tabla 2.34

Fuente: Grupo de Proyecto