



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Construcción de un sistema de
información para controlar el flujo
de mercancías a través de los
recintos fiscalizado**

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Alejandro Federico Tejeda Muñoz

DIRECTOR DE TESIS

M.I.A. Adolfo Millán Nájera



Ciudad Universitaria, Cd. Mx., 2004

*A mi madre Pilar
que con su amor, valores,
tenacidad y espíritu de superación
ha sido un ejemplo a seguir.*

*A mi esposa Mary
que con su apoyo, amor incondicional y
comprensión, ha sabido reorientar mi camino*

*A mi hija Estefanía
esa radiante luz que ha llegado
a iluminar mi vida y dar nueva
motivación a cada actividad que emprendo*

*A mis profesores
que con su maravillosa labor
lograron sembrar en mí
la semilla del conocimiento*

*Agradezco a mi director de tesis, el Maestro Adolfo Millán por el apoyo brindado para
hacer posible la realización de este trabajo.*

INDICE

	Página
Capítulo 1. Antecedentes	1
1.1 Definición de la situación actual	1
1.2 Justificación del Sistema	2
1.3 Objetivos y Alcance	3
Capítulo 2. Análisis	4
2.1 Requerimientos	4
2.1.1 Reglas del Negocio	5
2.1.2 Restricciones	9
2.1.3 Fuentes de Información, Interfaces y Usuarios	10
2.1.4 Productos	10
2.1.5 De Seguridad	11
2.2 Propuesta de Solución	14
2.2.1 Modelo de Datos	17
2.2.2 Modelo de Procesos	27
2.3 Modelo de Comportamiento	33
2.4 Prototipo	35
Capítulo 3. Diseño	40
3.1 Arquitectura del Sistema	40
3.1.1 Estructura de Datos	40
3.1.2 Estructuras del Sistema	48
3.2 Diseño de las Interfaces del Sistema	50
3.2.1 Diseño de las interfaces internas y externas	50
3.2.2 Diseño de las Interfaces de Usuario	59
3.3 Diseño Procedimental	61
3.4 Procedimientos de Seguridad y de Control	66
3.4.1 Seguridad	67
3.4.2 Control	73
3.5 Diseño del Plan de Pruebas	75
3.5.1 Casos de Prueba	76
3.5.2 Tipos de Pruebas	76
3.5.3 Niveles de Pruebas	78
3.6 Diseño del plan de transición	82
3.6.1 Estrategia de transición	82
3.6.2 Diseñar el plan de implantación del sistema	82
3.6.3 Diseñar el plan de capacitación	85
Capítulo 4. Construcción	86

Capítulo 5. Implantación	91
5.1 Pruebas de aceptación	91
5.2 Capacitación de los usuarios de la aplicación	92
5.3 Liberación y distribución de la aplicación	92
Conclusiones	94
Bibliografía	96

Capítulo 1. Antecedentes

Capítulo 1. Antecedentes

1.1 Definición de la situación actual

Los Recintos Fiscalizados son particulares facultados por la Secretaría de Hacienda para realizar indistintamente funciones de manejo, almacenaje, custodia, carga y descarga de las mercancías de comercio exterior.

Actualmente, el control de las mercancías que pasan a través de los Recintos Fiscalizados no se lleva de una manera automatizada, empleándose el siguiente procedimiento:

- El Recinto Fiscalizado ingresa a su almacén las mercancías que venían en un avión o en un embarque, después de cotejarlas contra lo que la aerolínea o la naviera declaró en los manifiestos de carga.
- Registra en su control las mercancías que ingresaron, el país de procedencia, la guía Master y House asignado por la aerolínea o por la empresa marítima, la cual ampara las mercancías; así como también las mercancías que son retiradas del almacén para su despacho a través de un agente aduanal, además de indicar si se ha retirado el total de la mercancía amparada en una guía o solo parte de ella.
- El Recinto Fiscalizado genera diariamente un reporte con todos los movimientos realizados durante el día, el cual se entrega al personal de la aduana encargado de llevar el control de los Recintos Fiscalizados. Así mismo, dentro de los primeros diez días de cada mes, genera un disquete conteniendo las mercancías que han causado abandono, es decir, aquellas mercancías que han permanecido en el almacén por un tiempo mayor a los plazos establecidos por la Ley Aduanera en su Artículo 29.
- El personal de la aduana transforma el archivo de los movimientos a formato de Excel, para posteriormente revisar su contenido.
- El encargado del control de los Recintos Fiscalizados, analiza el contenido de los archivos recibidos para darle seguimiento a las mercancías, es decir, elaborar la notificación de abandono, la cual debe ser entregada al destinatario de las mercancías y darle el seguimiento al abandono.

Las tareas de seguimiento para las mercancías en abandono son:

1. Registrar la fecha en que la mercancía causo abandono,
2. Notificar al destinatario o consignatario de las mercancías,
3. Registrar en que fecha se notificó al destinatario o consignatario.

4. A partir de la fecha en que se realizó la notificación de abandono comenzar a calcular el plazo establecido por la ley para las mercancías en abandono, dentro del cual el destinatario o consignatario debe de reclamar su mercancía (Art. 30 de la Ley Aduanera), en caso de no hacerlo durante el plazo establecido, se deberá generar un acuerdo de adjudicación, con el cual se notifica al Consejo Asesor que la mercancía pasó a ser propiedad del fisco y que debe acudir al Recinto Fiscalizado para retirar de éste la mercancía en cuestión.

1.2 Justificación del Sistema

Tomando en cuenta el incremento en las operaciones de comercio exterior y en el flujo de las mercancías que pasan a través de los Recintos Fiscalizados, es necesario implementar un nuevo esquema para el control de las mercancías, que permita:

- Reducir al mínimo la captura de información.

Para generar los archivos que alimentarán al sistema, se aprovecharán los sistemas de captura con los que cuentan los Recintos Fiscalizados para registrar el ingreso y salida de las mercancías a sus almacenes.

- Controlar de manera más eficiente y confiable las mercancías que ingresan a los Recintos Fiscalizados.

Al automatizar el proceso de recepción y validación de la información, no se requerirá de personal para realizarlo, reduciendo costo y tiempo, así mismo, al no existir intervención humana, la información intercambiada no puede ser alterada.

- Seguridad en el almacenamiento de la información.

La información se almacenará en equipos más robustos para los cuales ya se tienen definidos procesos de respaldo y recuperación, además de contar con la infraestructura que garantiza la continuidad en la operación de los mismos.

- Explotación de la información almacenada de acuerdo a las necesidades de los usuarios.

Al encontrarse la información almacenada en una base de datos, el usuario podrá solicitar cualquier explotación de la misma (reportes y/o consultas) conforme a sus requerimientos y en el momento que lo desee.

1.3 Objetivos y Alcance

Implantar un control que permita vigilar las actividades de los Recintos Fiscalizados adscritos a las Aduanas, asegurando que se cumplan las obligaciones que la Ley Aduanera establece para con ellos.

Su alcance abarca los medios de transporte marítimos y aéreos.

Capítulo 2. Análisis

Capítulo 1. Análisis

Identificar y entender cuáles son los requerimientos del sistema a desarrollar es de suma importancia, pues no importa que tan bien se diseñe y codifique, si éstos no son satisfechos, el sistema decepcionará al cliente al no obtenerse los resultados esperados.

El análisis de los requerimientos permite especificar el funcionamiento y desempeño del sistema, las interfaces con otros actores y establecer las restricciones que debe cumplir, además de permitir modelar el contenido de la información utilizada y producida por el sistema, y como es transformada por él.

Para alcanzar esta meta es necesario, antes que nada, entender el problema y alcance del sistema que se pretende realizar para resolverlo.

En el caso del problema que toca al presente trabajo se empleó la entrevista como medio para recabar los requerimientos del sistema y restricciones con las que debe de cumplir, también se realizó una investigación en la legislación vigente para complementarlas.

2.1 Requerimientos

Un requerimiento está definido como la condición o capacidad con la que un sistema debe cumplir. Se documentan las necesidades de información derivadas del análisis de la situación actual.

De las entrevistas realizadas con el usuario, se identificaron los siguientes requerimientos:

- Almacenar la información de las mercancías que ingresan al Recinto Fiscalizado, indicando si existen sobrantes o faltantes según sea el caso, cuando haya diferencias contra el manifiesto de carga.
- Registrar en el sistema las mercancías que el Recinto Fiscalizado entrega.
- Almacenar los movimientos de transferencia de mercancías entre almacenes.
- Calcular diariamente los plazos de abandono para las mercancías que aún se encuentran en los Recintos Fiscalizados y marcar en abandono aquellas a las que se les han vencido los plazos.
- Permitir al personal de la aduana consultar cuáles mercancías se encuentran en cada Recinto Fiscalizado, así como darle seguimiento a aquellas que han causado abandono.

2.1.1 Reglas del Negocio

Validación y Carga de Datos

El Recinto Fiscalizado (REFI) enviará los movimientos de mercancías (Ingresos, Notificaciones de Arribo y Salidas), los aprovechamientos y compensaciones del mes inmediato anterior, en archivos. Cada movimiento conformará un registro de acuerdo a las especificaciones del “Manual para la Interconexión Informática con los Recintos Fiscalizados Tráficos Aéreo, Marítimo y Terrestre”, proporcionado por la Administración General de Aduanas.

- a) La información enviada debe de cumplir con la estructura y formatos especificados en el manual citado para proceder a su validación.
- b) La validación de los registros de movimientos enviados por el REFI se hará cumpliendo las siguientes reglas para cada tipo de movimiento:

i) Ingreso de Mercancías

- (1) Inicia el ciclo al registrarse el ingreso de una guía al REFI, por lo que el cómputo para el plazo en que la guía causará abandono se origina a partir de la fecha que se declara que la guía ingresó al país.
- (2) La fecha de entrada declarada debe ser menor o igual a la fecha actual.
- (3) El tipo de ingreso declarado debe ser válido (según catálogo).
- (4) El tipo de mercancía declarado debe ser válido (según catálogo).
- (5) La unidad de medida declarada debe ser válida (según catálogo).
- (6) No se permite el ingreso por “Transferencia desde otro REFI” si no se ha registrado previamente en la base de datos, una salida por transferencia en el REFI origen hacia el de destino declarado.

ii) Notificación de Arribo de Mercancías

- (1) Debe existir en la base de datos un ingreso con la misma llave que la declarada en la notificación.
- (2) La fecha de notificación declarada debe ser mayor o igual a la fecha de entrada de la guía al país y menor o igual a la fecha actual.

iii) Salida de Mercancías

- (1) Debe existir en la base de datos un ingreso con la misma llave que la declarada en la salida.
- (2) La fecha de salida declarada debe ser mayor o igual a la fecha de entrada de la guía al país y menor o igual a la fecha actual.
- (3) La unidad de medida declarada debe ser válida (según catálogo).
- (4) Una guía puede ser retirada del REFI totalmente o por subdivisiones.
- (5) No se permite la salida por "Trasbordo" cuando ya se ha registrado en la base de datos, la salida de una subdivisión para esa guía.
- (6) No se permite la salida por "Transferencia hacia otro REFI" cuando no existe el REFI destino declarado.
- (7) Se debe retirar la guía completamente o lo que resta de ella, cuando se declara una salida por "Transferencia hacia otro REFI", es decir, ya no pueden salir más parcialidades o subdivisiones de la misma.
- (8) Cuando se ha retirado la totalidad de una guía del REFI se cierra el ciclo para ella, por lo cual ya no se puede registrar ningún otro movimiento sobre la misma.

iv) Aprovechamientos

- (1) El REFI declarado debe estar registrado y activo en la base de datos.
 - (2) No debe existir en la base de datos un aprovechamiento con la misma llave.
- c) Cada movimiento que cumpla con las reglas de validación será cargado en la base de datos.
- d) Se enviará al REFI el resultado de la validación en un archivo de respuesta, con las características especificadas en el "Manual para la Interconexión Informática con los Recintos Fiscalizados Tráficos Aéreo, Marítimo y Terrestre".

Abandono de Mercancías

De acuerdo con el Artículo 29 de la Ley Aduanera (La Ley) vigente:

“Causarán abandono en favor del Fisco Federal las mercancías que se encuentren en depósito ante la aduana en los siguientes casos:

- I. Expresamente, cuando los interesados así lo manifiesten por escrito.*
- II. Tácitamente, cuando no sean retirados dentro de los plazos que a continuación se indican:*
 - a) Tres meses, tratándose de la exportación.*
 - b) Tres días, tratándose de mercancías explosivas, inflamables, contaminantes, radiactivas o corrosivas, así como las mercancías perecederas o de fácil descomposición.*
 - c) Dos meses en los demás casos.*

También causarán abandono a favor del Fisco Federal las mercancías que hayan sido embargadas por las autoridades aduaneras con motivo de la tramitación de un procedimiento administrativo o judicial o cuando hayan sido vendidas o rematadas y no se retiren del recinto fiscal o fiscalizado. En estos casos causarán abandono en dos meses contados a partir de la fecha en que queden a disposición de los interesados.”

- e) El cómputo de los plazos iniciará al día siguiente en que se registró el ingreso de la guía al territorio nacional para la importación. Para la exportación será al día siguiente en que queden en depósito ante la aduana (Art. 30 de la Ley).*
- f) Los plazos a considerar para los abandonos serán conforme al catálogo proporcionado por la ADS que se muestra en la tabla 2.1.1:*

Clave	Descripción	Plazo en días hábiles
A	Animales vivos	10
B	Mercancías perecederas	5
C	Restos humanos	Importación - 60 Exportación - 90
D	Fármacos	Importación - 60 Exportación - 90

Clave	Descripción	Plazo en días hábiles
E	Materiales Impresos	Importación - 60 Exportación - 90
F	Calzado	Importación - 60 Exportación - 90
G	Eléctricos	Importación - 60 Exportación - 90
H	Equipo médico	Importación - 60 Exportación - 90
I	Muestras	Importación - 60 Exportación - 90
J	Partes y refacciones	Importación - 60 Exportación - 90
K	Electrónicos	Importación - 60 Exportación - 90
L	Textiles	Importación - 60 Exportación - 90
M	Comato (Company Materials)	Importación - 60 Exportación - 90
N	Comisariato (Utilería de Avión)	Importación - 60 Exportación - 90
O	Mercancías consolidadas	Importación - 60 Exportación - 90
S	Paquetería transportada por empresas de mensajería	Importación - 60 Exportación - 90
T	Otros materiales no peligrosos	Importación - 60 Exportación - 90
U	No especificados en la guía	Importación - 60 Exportación - 90
1	Mercancías explosivas	3
2	Gases comprimidos, licuados o disueltos a presión	3
3	Líquidos inflamables	3
4	Sólidos inflamables	3
5	Sustancias comburentes y peróxidos orgánicos	3
6	Sustancias venenosas, también denominadas tóxicas y sustancias infecciosas	3
7	Materiales radioactivos	3
8	Sustancias corrosivas	3
9	Sustancias peligrosas distintas a las señaladas anteriormente	3

tabla 2.1.1 Catálogo de tipos de mercancías y plazos

- g) De conformidad con el Artículo 32 de la Ley, cuando las mercancías hubieran causado abandono, las autoridades aduaneras notificarán con acuse de recibo a los propietarios o consignatarios de las mismas, que ha transcurrido el plazo de abandono y que cuentan con 15 días hábiles para retirarlas.

- h) Se computarán los 15 días hábiles citados en el inciso anterior, a partir de la fecha en que se realizó la notificación de abandono.
- i) De acuerdo con el Artículo 33,

“Los plazos de abandono se interrumpirán:

II. Por la interposición del recurso administrativo que corresponda conforme al Código Fiscal de la Federación o la presentación de la demanda en el juicio que proceda.

El recurso o la demanda solo interrumpirán los plazos de que se trata, cuando la resolución definitiva que recaiga no confirme, en todo o en parte, la que se impugnó.”

Mercancías a Disposición del Consejo Asesor

Serán mercancías a disposición del Consejo Asesor cuando se ha cumplido el plazo de 15 días hábiles, citado en el Artículo 32 de la Ley.

- j) Se notificará con acuse de recibo a los propietarios y consignatarios de las mercancías el acuerdo administrativo mediante el cual se determina que las mercancías han pasado a propiedad del Fisco Federal.
- k) Para que el Consejo Asesor pueda disponer de la mercancía debe cumplirse un plazo de 55 días hábiles, después del cual el Consejo Asesor contará con 45 días hábiles para poderla retirar del REFI.

2.1.2 Restricciones

Dentro de las restricciones establecidas por la Administración Pública Federal se encuentra la de austeridad, la cual implica hacer uso más racional de los recursos (materiales, humano, de equipamiento, etcétera) y aprovechar al máximo los existentes.

Por lo anterior, el sistema a desarrollar deberá aprovechar la infraestructura actualmente instalada por lo que no será necesario definirla ni realizar las actividades requeridas para la evaluación de diferentes propuestas tecnológicas para seleccionar la más apropiada.

Las aduanas actualmente cuentan con servidores HP9000 con UNIX HP-UX v.10.0, INFORMIX como manejador de base de datos relacional, así mismo, las áreas administrativas cuentan con PC's con Windows como sistema operativo y un servidor con Windows NT v.3.

Una restricción establecida por la Administración de Desarrollo, es que no se podrá emplear un manejador de ODBC (Open Data Base Connectivity) para acceder a los datos del servidor desde el cliente, al no contarse con una herramienta comercial para el control de accesos.

Herramientas de programación

La institución cuenta con licencias institucionales para herramientas de desarrollo de las cuales se seleccionaron las siguientes para la construcción del sistema:

Para la plataforma UNIX:

- ANSI C y Embedded C (ESQL/C) para INFORMIX

Para la plataforma Intel (Windows)

- Visual Basic y Visual C++.

2.1.3 Fuentes de Información, Interfaces y Usuarios

Se documentan los reportes, formatos, comunicados, etcétera, que integran los elementos de información que el área requiere para operar, así como los que interactúan con otros sistemas y los requeridos por el usuario.

2.1.4 Productos

Se debe definir claramente que salidas espera el cliente del sistema, como son los reportes, consultas y otras salidas con relación al producto. Lo cual permitirá dimensionar el sistema y definir los módulos requeridos para ello.

Los productos requeridos por el usuario son:

Consultas

Módulo de consulta de las diferentes etapas que las mercancías atraviesan en su paso por los REFIS (Recintos Fiscalizados), el cual debe permitir llevar a cabo las consultas para todos los REFIS a la vez o alguno en específico, durante un período determinado por el usuario.

- **Entradas.**- Consulta de la entrada de mercancías registradas en cada REFI.

- **Notificación de Arribo.-** Consulta de las notificaciones realizadas por los REFIS a los destinatarios con respecto al arribo de las mercancías.
- **Salidas.-** Consulta de la salida de mercancías registradas en cada REFI.
- **Mercancías.-** Consulta de las mercancías que se encuentran en los REFIS, clasificándolas de la siguiente forma:

Peligrosas
Destruídas
Extraviadas
Exentas de Abandono
En Abandono
En Controversia
Adjudicadas
A Disposición del Consejo Asesor
Retiradas por el Consejo Asesor

- **Aprovechamientos.-** Consulta de los ingresos que reportan los REFIS por concepto de almacenaje.

Reportes

Módulo de emisión de listados sobre las diferentes etapas que las mercancías atraviesan en su paso por los REFIS (Recintos Fiscalizados), permitiendo generar reportes para todos los REFIS a la vez o alguno en específico, durante un período determinado por el usuario, además de permitirle imprimir los campos que éste seleccione y en el orden deseado para los mismos conceptos definidos para las consultas.

2.1.5 De Seguridad

Es importante considerar los requerimientos de seguridad para el sistema a desarrollar con la finalidad de cumplir con las normas y políticas establecidas por el cliente, en caso de no existir, definir las para garantizar la integridad en la información y restringir el acceso sólo al personal o áreas autorizadas.

Para lo cual se consideran los siguientes puntos:

Clasificación de la información

En el caso de la SHCP, existen políticas de seguridad ya definidas para la clasificación de la información con base en:

1) **Confidencialidad.**

Se debe indicar el grado de importancia de la información manejada, considerando que no sea permitido que ésta se conozca y cuál de ella por su bajo nivel de confidencialidad es permitido que todo público la conozca.

Confidencial restringido. Aplica para la información que, si se usa o divulga de una manera inapropiada, podría perjudicar a los intereses del SAT o del gobierno.

Información confidencial. Aplica para la información que debe estar protegida contra la divulgación no autorizada. Esto incluye la información de naturaleza personal, detalles privados de la operación de una empresa, compañía o comercio con el cual tiene contacto el SAT, o detalles privados de la operación de la organización.

Información de uso interno. Aplica para la información que no está considerada como confidencial pero que no debe distribuirse fuera del SAT.
Información Pública. Aplica para la información que no se considera sensible para las operaciones del SAT y que no está sujeta a leyes nacionales en relación con la privacidad de los clientes del SAT.

2) **Integridad.**

Indicar qué tan importante es evitar que la información sea alterada de su estado original, a tal grado que la operación se ponga en riesgo por el simple hecho de que exista un cambio no previsto o permitido.

Alto. Si el hecho de modificar la integridad del activo de información, resulta desastroso para la operación de la Institución.

Medio. Aplica para aquella información que aún modificada, es posible continuar con la operación del sistema o aplicación.

Bajo. Representa el nivel de importancia más baja en caso de alguna alteración en el contenido de la información, puesto que ésta no pone en riesgo la operación de la aplicación o sistema y que pueda ser, por ejemplo, regenerada mediante un respaldo.

3) Disponibilidad.

Establecer la importancia de que la información se encuentre disponible continuamente, evitando con esto el poner en riesgo la operación de la aplicación o sistema, a partir de una interrupción en el servicio.

Alto. Si el hecho de que la información no se encuentre disponible para la continuidad de la operación, afecta directamente a los intereses de la Institución y el servicio ya sea para el contribuyente o el personal interno.

Medio. Aplica para aquella información que si no se encuentra disponible por un lapso relativo de tiempo, ésta no repercute en la operación de la aplicación o sistema al que pertenece.

Bajo. Este es el nivel más bajo de importancia, el cual indica claramente que no importa que la información no se encuentre disponible o bien que el sistema se encuentre fuera de operación por un lapso relativo de tiempo.

Con base en lo anterior, se clasificó la información manejada por el sistema como se muestra en la tabla 2.1.2:

Descripción	Medio	Sensibilidad		
		Confidencialidad	Integridad	Disponibilidad
Archivo de Movimientos	E-MAIL	Confidencial	Alta	Alta
Base de Datos	DBSpace	Confidencial Restringido	Alta	Alta
Consultas	Front-End	Confidencial Restringido	Baja	Alta
Reporte	Impresión	Confidencial Restringido	Baja	Alta

tabla 2.1.2 Clasificación de la mercancía

Acceso lógico

Todo sistema o aplicación debe contar con un aceptable esquema de control de acceso y considerar los tres siguientes aspectos elementales:

Control de acceso al sistema

1) Autorización

El sistema debe considerar una opción para registrar a los usuarios que serán autorizados para su utilización y los recintos fiscalizados con los cuales se intercambiará información.

2) Autenticación

El sistema debe contar con los elementos que le permitan garantizar la autenticidad del usuario que pretende acceder al sistema, así como la procedencia de la información enviada por los recintos fiscalizados.

3) Revocación

El sistema debe permitir revocar ya sea temporal o definitivamente el acceso al sistema a cualquier usuario cuando así se requiera.

Encriptación

El concepto de encriptación es empleado con mayor fuerza hoy en día principalmente en la transferencia de información, sin embargo es igual de importante considerar la encriptación en toda aquella información que es considerada como sensible.

1. El uso o no de encriptación en la aplicación, depende directamente de la clasificación de la información, por lo que toda información considerada como sensible y de uso confidencial deberá ser manejada mediante cualquier mecanismo autorizado de encriptación.
2. Las contraseñas deberán ser manejadas en forma muy segura dentro de la aplicación, contemplando métodos de encriptación para evitar la posible violación de información.
3. La transferencia de información entre equipos considerada como sensible, debe considerar alguno de los métodos de encriptación.

Considerando que la información a ser enviada por los Recintos Fiscalizados es sensible, será necesario emplear algún mecanismo de encriptación para garantizar la confidencialidad e integridad de la información.

2.2 Propuesta de Solución

Se efectúa en forma preliminar el modelado y prototipo del sistema propuesto que representa la solución a los requerimientos planteados por el usuario.

A continuación se listan diferentes alternativas de solución propuestas:

1. Que los **REFIS** establezcan una conexión directa con el Servidor HP9000 para entregar y recibir archivos. La transferencia de los archivos se hace empleando Sockets de Berkeley bajo TCP/IP.

La validación de los archivos, así como la carga de información se hace en el Servidor HP9000 aprovechando el Manejador de Informix.

Al personal encargado del Control de los **REFIS** se le asigna una terminal para explotar la información almacenada, así como la actualización de la misma.

Los Notificaciones de Abandono y Acuerdos de Adjudicación se imprimen en la impresora conectada al Servidor HP9000.

2. Que los **REFIS** establezcan una conexión directa con el Servidor HP9000 para entregar y recibir archivos. La transferencia de los archivos se hace empleando Sockets de Berkeley bajo TCP/IP.

La validación de los archivos, así como la carga de información se hace en el Servidor HP9000 aprovechando el Manejador de Informix.

Al personal encargado del Control de los **REFIS** se les asigna una PC para explotar la información almacenada, así como la actualización de la misma.

Los Notificaciones de Abandono y Acuerdos de Adjudicación se imprimen en alguna de las impresoras laser conectadas al Servidor de Impresión.

3. Los **REFIS** entregan y reciben sus archivos en disquete.

Los archivos de movimientos se descargan del disquete a una PC por personal del Control de **REFIS**.

De la PC se transfieren los archivos hacia el Servidor HP9000 para su validación y carga.

El Servidor HP9000 transmite el archivo de respuesta a la PC.

La PC carga en el disquete los archivos de respuesta. El disquete es entregado al REFI.

Al personal encargado del Control de los **REFIS** se les asigna una PC para explotar la información almacenada, así como la actualización de la misma.

Los Notificaciones de Abandono y Acuerdos de Adjudicación se imprimen en alguna de las impresoras laser conectadas al Servidor de Impresión.

4. Los **REFIS** entregan y reciben sus archivos a través de Correo Electrónico.

Uno de los Servidores NT que controlan los accesos, se encarga de descargar automáticamente del buzón de correo electrónico los archivos de movimientos.

Del Servidor NT se transfieren los archivos hacia el Servidor HP9000 para su validación y carga.

El Servidor HP9000 transmite el archivo de respuesta al Servidor NT.

El Servidor NT envía a los **REFIS** los archivos de respuesta a través del Correo Electrónico por Internet.

Al personal encargado del Control de los **REFIS** se les asigna una PC para explotar la información almacenada, así como la actualización de la misma.

Los Notificaciones de Abandono y Acuerdos de Adjudicación se imprimen en alguna de las impresoras laser conectadas al Servidor de Impresión.

Solución

Se opta por alternativa de solución número cuatro por lo siguiente:

- Las alternativas uno y dos involucran una inversión para los **REFIS** tanto en infraestructura, para poder conectarse al Servidor HP9000 de la aduana, así como para el desarrollo de los sockets de transmisión y recepción de archivos.
- En la alternativa uno se proponen terminales para hacer la interfaz con el usuario, las cuales son en modo texto con las limitaciones que esto implica.
- La alternativa tres requiere de recurso humano para la recepción y entrega de disquetes tanto por los **REFIS** como por la aduana, aunque elimina el costo en infraestructura y desarrollo, también depende del horario de operación.
- En la alternativa cuatro, los **REFIS** solo requieren contratar un servicio de internet y una cuenta de correo electrónico para transferir y recibir archivos en el momento que ellos lo deseen, reduciéndose sustancialmente la inversión en cuanto a recursos materiales y tiempo. Esta alternativa se podría complementar con la tres, en cuanto el empleo de disquetes en caso de una contingencia.

La propuesta de solución para el caso se basa en un esquema Cliente-Servidor, en el cual los Recintos Fiscalizados transmitirán archivos conteniendo la información de los movimientos realizados. Dichos archivos serán procesados por lotes (*batch*). Una vez procesados por el sistema, éste devolverá a los Recintos Fiscalizados archivos con el resultado del proceso. La transmisión de los archivos de movimientos y de respuesta se hará empleando correo electrónico (e-mail).

Por otro lado, el personal de la aduana podrá consultar y generar reportes de la información proporcionada por los Recintos Fiscalizados, además de dar seguimiento a las mercancías en abandono. La figura 2.2.1 muestra el diagrama conceptual de la solución propuesta.

En el desarrollo de la propuesta que dará solución a los requerimientos, se plasmarán los datos a ser manejados por el sistema, así como los procesos que los transforman, elaborándose los modelos que la metodología para análisis y diseño estructurado de datos proporcionan.

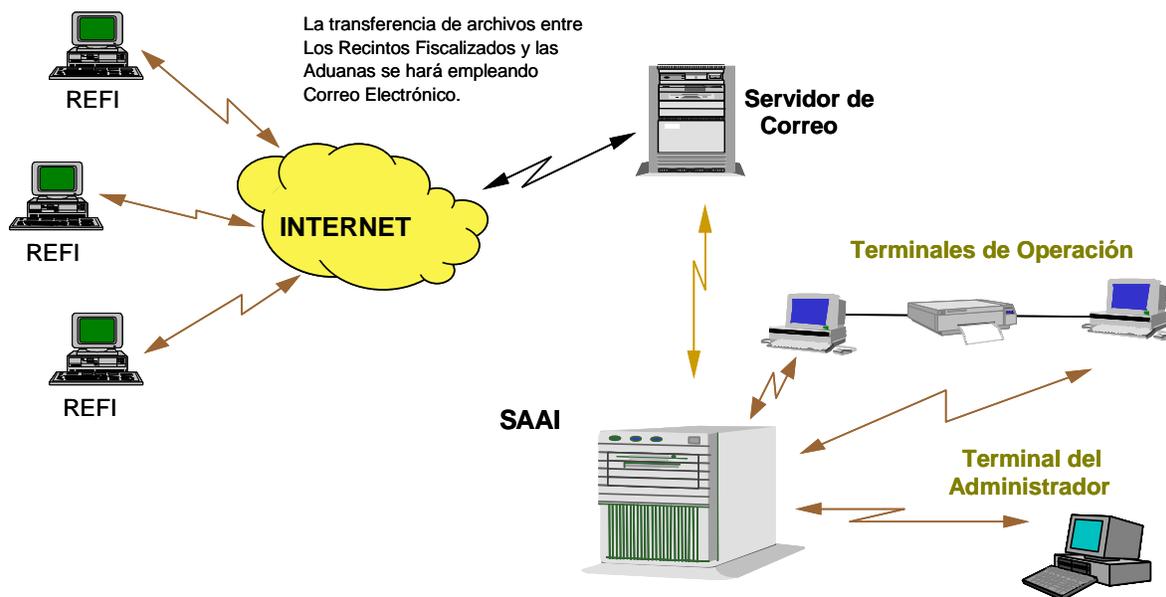


figura 2.2.1 Diagrama Conceptual.

2.2.1 Modelo de Datos

El modelo de datos nos permitirá representar la información que es procesada por el sistema (Dominio de Datos), para ello es importante identificar las entidades que intervienen, así como su ámbito de acción para poder clasificarlas como externas o internas; también permite identificar cuáles son los objetos de datos a ser procesados, cómo están compuestos, qué atributos los definen y qué relación tienen cada uno de ellos entre sí.

El modelo de datos consiste de tres partes, los objetos de datos (entidades), los atributos que describen a cada objeto o entidad, y las relaciones que conectan a los objetos entre sí.

Una entidad es un objeto, cosa real o abstracta con varias propiedades o atributos comunes entre sí. Por ejemplo un auto el cual puede tener los siguientes atributos: marca, modelo, VIN, color, etc.

Los atributos definen las propiedades del objeto o entidad. Uno o más atributos pueden identificar a las entidades, a éstos se les llaman llaves, en ocasiones los valores que estos atributos pueden tomar permiten identificar de manera única a las entidades, en el caso de un auto el VIN (número de identificación vehicular, de sus siglas en inglés vehicle identification number) lo identifica de manera única ya que no existe otro igual.

Es importante poder distinguir las entidades de los atributos por ejemplo: Ancho no es una entidad sino un atributo, pero Dimensiones sí lo puede ser ya que está compuesto de ancho, alto, largo, peso.

Una relación es la conexión o dependencia que existe entre objetos o entidades determinada por el papel que juegan cada uno de ellos dentro de nuestro sistema, por ejemplo, consideremos las entidades libro y librería, algunas de las relaciones que existen entre ellas pueden ser:

- Una librería vende libros
- Una librería almacena libros
- Una librería exhibe libros

Lo anterior se puede representar gráficamente como se muestra en la figura 2.2.2.

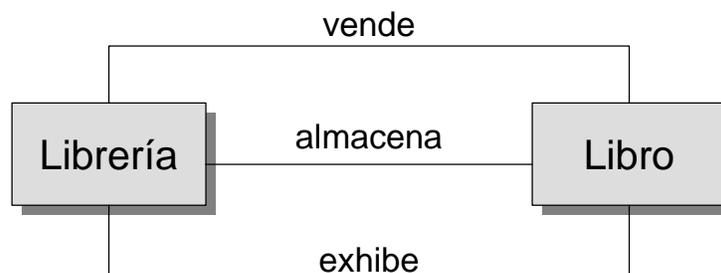


figura 2.2.2 Relación entre entidades

Es importante hacer notar que las relaciones son bidireccionales y pueden ser leídas en ambos sentidos. La librería vende libros, o bien, un libro es vendido por una librería.

Las relaciones nos representan las dependencias que pueden existir entre un par de objetos, pero el modelo de datos debe además proporcionar información de cuántas ocurrencias de cada objeto existen en dicha relación, lo que nos conlleva

al concepto de cardinalidad el cual nos especifica qué número de ocurrencias de un objeto se relacionan con el número de ocurrencias de otro.

La cardinalidad puede ser:

- Uno a uno (1:1) – Una ocurrencia del objeto 'X' puede relacionarse con una y solo una ocurrencia del objeto 'Y' y viceversa, por ejemplo en el caso de México, un esposo puede tener sólo una esposa y una esposa sólo un esposo.
- Uno a muchos (1:n) – Una ocurrencia del objeto 'X' puede relacionarse con una o muchas ocurrencias del objeto 'Y', pero una ocurrencia del objeto 'Y' con tan solo una del objeto 'X', por ejemplo una madre puede tener muchos hijos, pero un hijo solo una madre.
- Muchos a muchos (m:n) – Una ocurrencia del objeto 'X' puede relacionarse con una o muchas ocurrencias del objeto 'Y', y una ocurrencia del objeto 'Y' con una o muchas del objeto 'X', por ejemplo un tío puede tener muchos sobrinos y un sobrino puede tener muchos tíos.

El gráfico que nos permite representar las entidades y las relaciones existentes entre ellos, se conoce como Diagrama Entidad – Relación.

Para el problema que se está desarrollando se detectaron las siguientes entidades con sus atributos como se muestran en la tabla 2.2.3 a la tabla 2.2.17.

Entidad	Aduana
Descripción	Catálogo de Aduanas
Atributo	Descripción
Clave de la Aduana	Clave de la Aduana.
Nombre	Nombre de la aduana.

tabla 2.2.3 Entidad Aduana

Entidad	Aprovechamiento
Descripción	Montos de aprovechamiento mensual declarados por cada Recinto Fiscalizado.
Atributo	Descripción
Clave del Recinto Fiscalizado	Clave del Recinto Fiscalizado
Clave de la Aduana	Clave de la Aduana.
Mes	Mes que se declara.
Año	Año que se declara.
Ingreso Bruto	Ingreso bruto durante el mes declarado.
Monto compensable de Inversión	Monto compensable por inversión en el mes declarado.
Monto Almacenaje	Monto compensable por almacenaje en el mes declarado.

Monto Pagado	Importe del aprovechamiento en el mes declarado.
--------------	--

tabla 2.2.4 Entidad Aprovechamiento

Entidad	Extracción
Descripción	Guías retiradas del Recinto Fiscalizado.
Atributo	Descripción
Agente Aduanal	Agente Aduanal que tramitó el pedimento con el cual se extrae la mercancía.
Número de Pedimento	Número de pedimento con el cual se extrae la mercancía.
Aduana Destino	Aduana en la que se ubica el Recinto Fiscalizado al que se dirige la mercancía en caso de una transferencia.
Recinto Destino	Recinto Fiscalizado al que se dirige la mercancía en caso de una transferencia.
Cantidad Extraída	Cantidad de mercancía extraída en unidades de medida
Folio Acta Extracción	Folio del Acta con la que se extrae la mercancía.
Clave Unidad Medida	Clave de la unidad de medida.
Cantidad de Merma	Cantidad de mercancía faltante en unidades de medida
Clave de la Aduana	Clave de la Aduana.
Registro de Entrada	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
Clave del Recinto	Clave del Recinto Fiscalizado
Fecha de Extracción	Fecha en que extrae la mercancía del Recinto Fiscalizado.
Clave Tipo Salida	Clave del tipo de salida o extracción.
Fecha de Entrada	Fecha en que ingresó la guía al Recinto Fiscalizado
Bandera Última	Bandera que indica si la subdivisión extraída es la última.
Número de Subdivisión	Número de subdivisión extraída.

tabla 2.2.5 Entidad Extracción

Entidad	Guía
Descripción	Guías que ingresan a los Recintos Fiscalizados
Atributo	Descripción
Clave del Recinto	Clave del Recinto Fiscalizado
Clave de la Aduana	Clave de la Aduana.
Registro de Entrada	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.

Fecha de Entrada	Fecha en que ingresó la guía al Recinto Fiscalizado
Clave del Tipo Entrada	Clave del tipo de entrada
Tipo Guía	Clave del tipo de guía
Clave Tipo Mercancía	Clave del tipo de mercancía.
Total de piezas	Total de piezas contenidas en la guía.
Clave Unidad Medida	Clave de la unidad de medida.
Cantidad	Cantidad de mercancía en unidades de medida
Guía Master	Número de la guía master.
Guía house	Número de la guía house.
Folio Acta	Folio del Acta con la ingresa la mercancía al Recinto Fiscalizado.
Costo Diario de Almacenaje	Costo diario promedio de almacenaje que causa la mercancía.
Recinto Origen	Clave del Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
Aduana Origen	Clave de la aduana donde se ubica el Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
Registro de entrada original	Número de Registro asignado por Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
Fecha de Entrada Original	Fecha en que ingreso la mercancía al país en caso de ingreso por transferencia.
Agente Aduanal	Patente del Agente Aduanal que tramitó el pedimento con el cual ingresó la mercancía al Recinto Fiscalizado.
Número de pedimento	Número de pedimento con el cual ingresó la mercancía al Recinto Fiscalizado.
Registro Federal de Causante	Registro Federal de Causante del Destinatario/Consignatario de las mercancías.
Nombre del Destinatario	Nombre o Razón Social del Destinatario/Consignatario de las mercancías.
Tipo domicilio	Tipo de domicilio del Destinatario/Consignatario de las mercancías.
Folio Irregularidades	Folio del Acta en la cual se registraron las irregularidades encontradas.
Estado	Estado en el que se encuentra la guía.

tabla 2.2.6 Entidad Guía

Entidad	Guía en Abandono
Descripción	Guías en abandono
Atributo	Descripción
Clave del Recinto	Clave del Recinto Fiscalizado

Clave de la Aduana	Clave de la Aduana.
Registro de Entrada	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
Fecha de Entrada	Fecha en que ingresó la guía al Recinto Fiscalizado
Fecha de Abandono	Fecha en que la mercancía causó abandono.
Operador	Operador que generó la notificación de abandono.
Fecha de Generación	Fecha en que se generó la notificación de abandono.
Folio Notificación	Folio de la notificación.
Notificador	Número de gafete del Notificador
Fecha de Notificación	Fecha en que se notificó del abandono al Destinatario/Consignatario.
Fecha Controversia	Fecha en la que el Destinatario/Consignatario promovió una controversia.
Fecha Resolución	Fecha en que se dictó la resolución a la controversia promovida por el Destinatario/Consignatario.
Instancia	Instancia ante la que se promovió la controversia.
Folio Resolución	Folio asignado por la instancia a la resolución emitida por ella.
Estado	Estado en el que se encuentra el abandono.

tabla 2.2.7 Entidad Guías en abandono

Entidad	Notificación de Arribo
Descripción	Control de las notificaciones de arribo hechas a los destinatarios/consignatarios de las mercancías.
Atributo	Descripción
Código Postal	Código Postal donde se ubica el domicilio del destinatario/consignatario de las mercancías.
Colonia del Destinatario	Colonia donde se ubica el domicilio del destinatario/consignatario de las mercancías.
Calle del Destinatario	Calle donde se ubica el domicilio del destinatario/consignatario de las mercancías.
Ciudad del Destinatario	Ciudad donde se ubica el domicilio del destinatario/consignatario de las mercancías.
Estado del Destinatario	Estado donde se ubica el domicilio del destinatario/consignatario de las mercancías.
País del Destinatario	Clave del País
Clave de la Aduana	Clave de la Aduana.
Clave del Recinto	Clave del Recinto Fiscalizado
Tipo de Aviso	Cómo se notificó al destinatario/consignatario

	que las mercancías arribaron al Recinto Fiscalizado.
Fecha de Notificación	Fecha en que se notificó al destinatario/consignatario que las mercancías arribaron al Recinto Fiscalizado.
Registro de Entrada	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
Fecha de Entrada	Fecha en que ingresó la guía al Recinto Fiscalizado

tabla 2.2.8 Entidad Notificación de arribo

Entidad	Operador
Descripción	Catálogo de los operadores del sistema.
Atributo	Descripción
Clave del Operador	Clave del Operador
Clave del Puesto	Clave del puesto.
Gafete	Número de gafete asignado al operador.
Registro Federal de Causante	Registro Federal de Causantes del operador.
Nombre	Nombre del operador.
Contraseña	Contraseña de acceso al sistema.

tabla 2.2.9 Entidad Operador

Entidad	País
Descripción	Catálogo de países.
Atributo	Descripción
Clave del País	Clave del País
Nombre	Nombre del País

tabla 2.2.10 Entidad País

Entidad	Puesto
Descripción	Catálogo de puestos.
Atributo	Descripción
Clave del puesto	Clave del puesto.
Descripción	Descripción del puesto.

tabla 2.2.11 Entidad Puesto

Entidad	Recinto Fiscalizado
Descripción	Catálogo de Recintos Fiscalizados
Atributo	Descripción
Llave	Llave
Dirección E-Mail	Dirección de correo electrónico.
Fecha de Alta	Fecha en que se dio de alta el Recinto Fiscalizado.
Fecha de Baja	
Clave del Recinto	Clave del Recinto Fiscalizado

Domicilio	Domicilio del Recinto Fiscalizado.
Clave de la Aduana	Clave de la Aduana.
Nombre	Nombre del Recinto Fiscalizado.

tabla 2.2.12 Entidad Recinto Fiscalizado

Entidad	Recurso
Descripción	Recursos del sistema a los que tiene acceso el operador
Atributo	Descripción
Clave del Operador	Clave del Operador
Recurso	Recurso al que se tiene acceso

tabla 2.2.13 Entidad Recurso

Entidad	Tipo de Entrada
Descripción	Catálogo del tipo de entradas
Atributo	Descripción
Clave del Tipo Entrada	Clave del tipo de entrada
Descripción	Descripción del tipo de entrada

tabla 2.2.14 Entidad Entrada

Entidad	Tipo de Mercancía
Descripción	Catálogo del Tipo de Mercancías
Atributo	Descripción
Clave Tipo de Mercancía	Clave del tipo de mercancía.
Descripción	Descripción del tipo de mercancía.
Plazo	Plazo en días para retirar la mercancía del Recinto Fiscalizado.
Bandera Peligrosa	Bandera de mercancía peligrosa 0 = No Peligrosa. 1 = Peligrosa.

tabla 2.2.15 Entidad Tipo de mercancía

Entidad	Tipo de Salida
Descripción	Catálogo del Tipo de Salidas
Atributo	Descripción
Clave del Tipo de Salida	Clave del tipo de salida o extracción.
Descripción	Descripción del tipo de salida o extracción.

tabla 2.2.16 Entidad Tipo de salida

Entidad	Unidad de Medida
Descripción	Catálogo de unidades de medida.
Atributo	Descripción
Clave Unidad Medida	Clave de la unidad de medida.
Descripción	Descripción de la unidad de medida.

tabla 2.2.17 *Entidad Unidad de medida*

En la figura 2.2.18 se muestra el Diagrama Entidad Relación obtenido para el sistema.

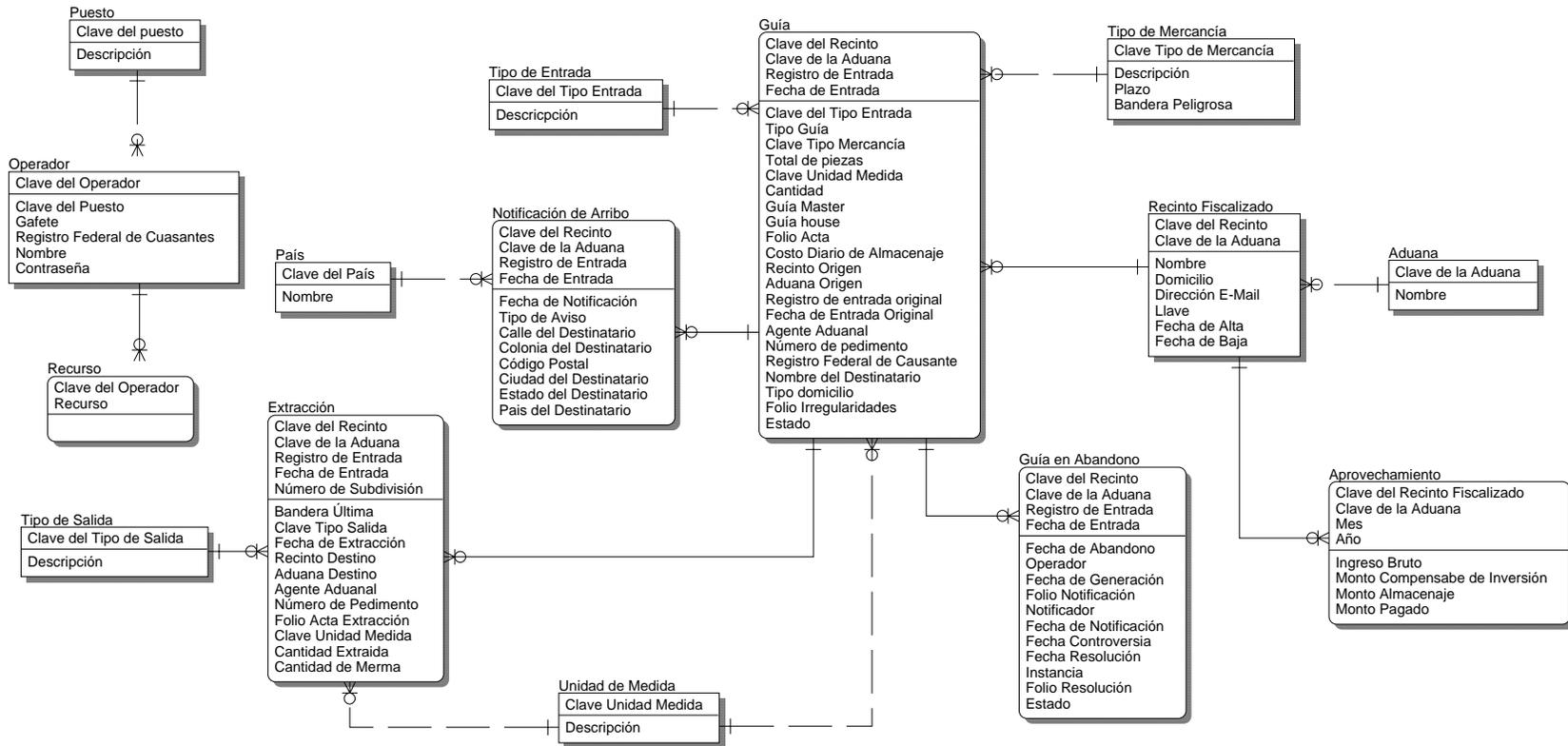


figura 2.2.18 Diagrama Entidad - Relación (ERD)

2.2.2 Modelo de Procesos

Además de los objetos de datos es necesario poder modelar los procesos que se encargan de transformar la información que fluye a lo largo del sistema (Dominio Funcional).

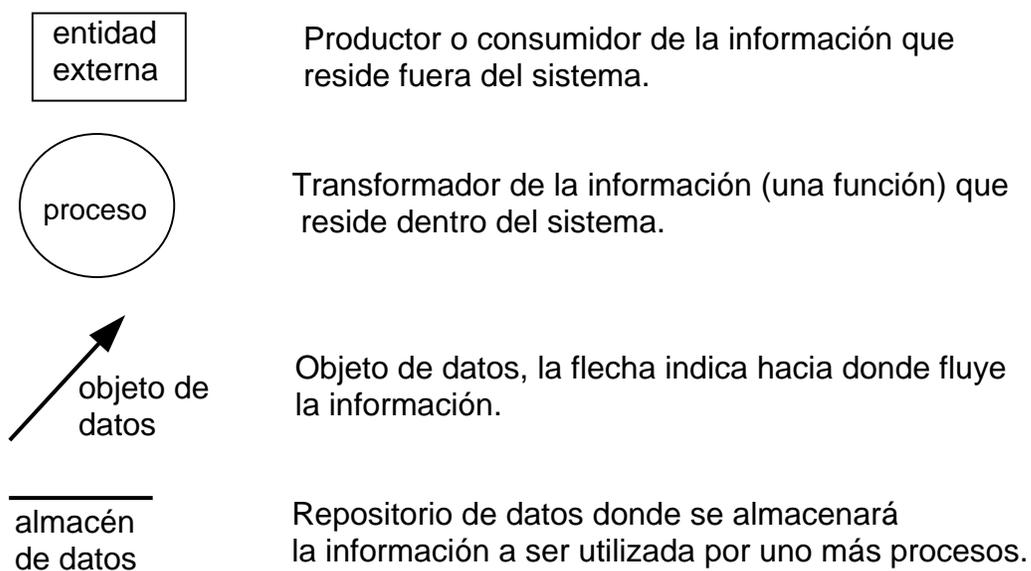
La información suministrada al sistema (entrada) puede ser de índole diversa y ser proporcionada por diferentes entidades externas al sistema, por ejemplo una señal de control enviada por un transductor, una cadena de caracteres suministrados por un capturista, una señal de audio captada por un micrófono, etcétera.

El sistema puede realizar diferentes procesos de transformación a la información recibida y entregar los resultados (salidas). Los procesos de transformación pueden involucrar cálculos numéricos muy complejos o simplemente realizar una comparación entre dos datos, así mismo, las salidas pueden ser, desde encender un led hasta modificar la trayectoria de un satélite.

El Diagrama de Flujo de Datos (DFD) o diagrama de burbujas, es el gráfico que nos permite modelar la información que fluye a través del sistema y los procesos transformadores que son aplicados a las entradas para obtener las salidas, así como las entidades que las suministran y consumen.

El DFD puede ser dividido en varios niveles para ahondar en los detalles de funcionalidad. El nivel 0 de un DFD es conocido también como diagrama de contexto, el cual representa a todo el sistema como una burbuja, con sus entradas y salidas indicadas por flechas entrantes y salientes respectivamente, para representar mayor detalle es necesario particionar el DFD de nivel 0.

A continuación se muestra la notación básica empleada en un DFD.



La figura 2.2.19 muestra el DFD de Nivel 0 ó diagrama de contexto para el sistema propuesto.

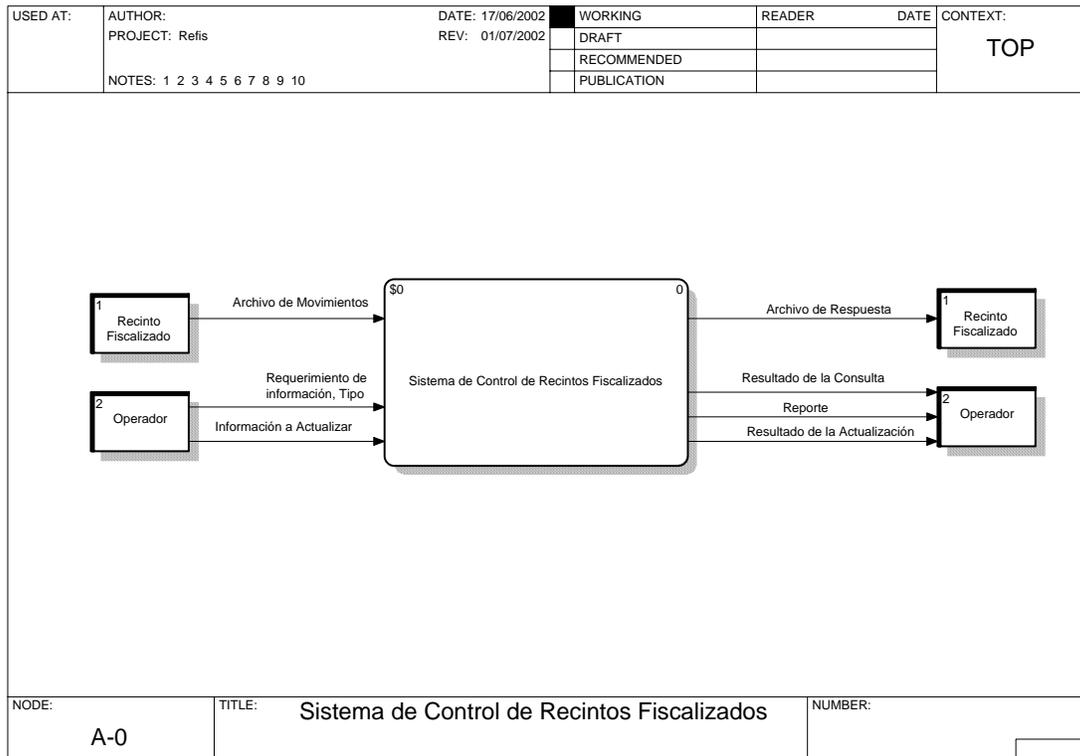


figura 2.2.19 Diagrama de Contexto

Nota.- En este caso debido a la herramienta CASE empleada para la elaboración de los modelos, se cambia la simbología para procesos, se emplea en lugar de la cual también es válida.

Descomponiéndolo para visualizar los procesos que lo conforman, obtenemos el diagrama de la figura 2.2.20, en el cual se representan los cuatro procesos principales que conforman al sistema:

- **Administrador de Archivos.** Se encargará de extraer de la bandeja de entrada los mensajes que contienen los archivos de movimientos transmitidos por los Recintos Fiscalizados, una vez extraído, el archivo es transmitido al Servidor de la Aduana, terminada la transmisión, es entregado al Validador para que verifique que la información contenida en ellos cumpla con la definición de los registros (validación sintáctica y catalógica) y con la normatividad vigente (validación normativa). Al concluir la validación de los archivos, se genera un archivo de respuesta, el cual es transmitido al servidor de correo para que se deposite en la bandeja de salida con la dirección de correo electrónico del Recinto Fiscalizado correspondiente.

Es necesario expandir este módulo para poder representar en el modelo lo que se plasmó en el diagrama conceptual (figura 2.2.20) y llegar a un nivel adecuado de detalle. La figura 2.2.21, muestra el diagrama de Nivel 2 obtenido al expandir este módulo.

- **Validador.** Este módulo se encargará de comprobar que la información contenida en los archivos de movimientos cumpla con los siguientes puntos:
 - a. Validación Sintáctica-Catalógica. Que los registros declarados dentro del archivo cumplan con la definición de los registros en cuanto a la sintaxis y tipo de dato, así mismo, que los datos que pertenecen a un catálogo existan en el mismo.
 - b. Validación Normativa. Que la información declarada en cada registro cumpla con la normatividad vigente, en este caso con la Ley Aduanera.

Cuando la información cubrió correctamente los puntos anteriores, éstos son almacenados en la base y se escribe un acuse en el archivo de respuesta, en caso contrario, se rechaza la información y se genera un código de error indicando la causa del rechazo, el cual se escribe en el archivo (figura 2.2.22 y figura 2.2.23).

- **Generación de Abandonos.** Este proceso se encargará de obtener las mercancías que han causado abandono al iniciar el día y de registrarlas en la tabla de abandonos para que el usuario les dé seguimiento, así mismo, buscará aquellas mercancías en abandono que ya fueron notificadas y que pasados quince días no han sido extraídas del Recinto Fiscalizado (figura 2.2.24).
- **Explotación de Información.** A través de este módulo, el usuario podrá consultar la información que se encuentra almacenada en la base de datos y darle seguimiento a las mercancías en abandono e imprimir las notificaciones de abandono, además de darle el mantenimiento necesario a los catálogos empleados por el sistema (figura 2.2.25).

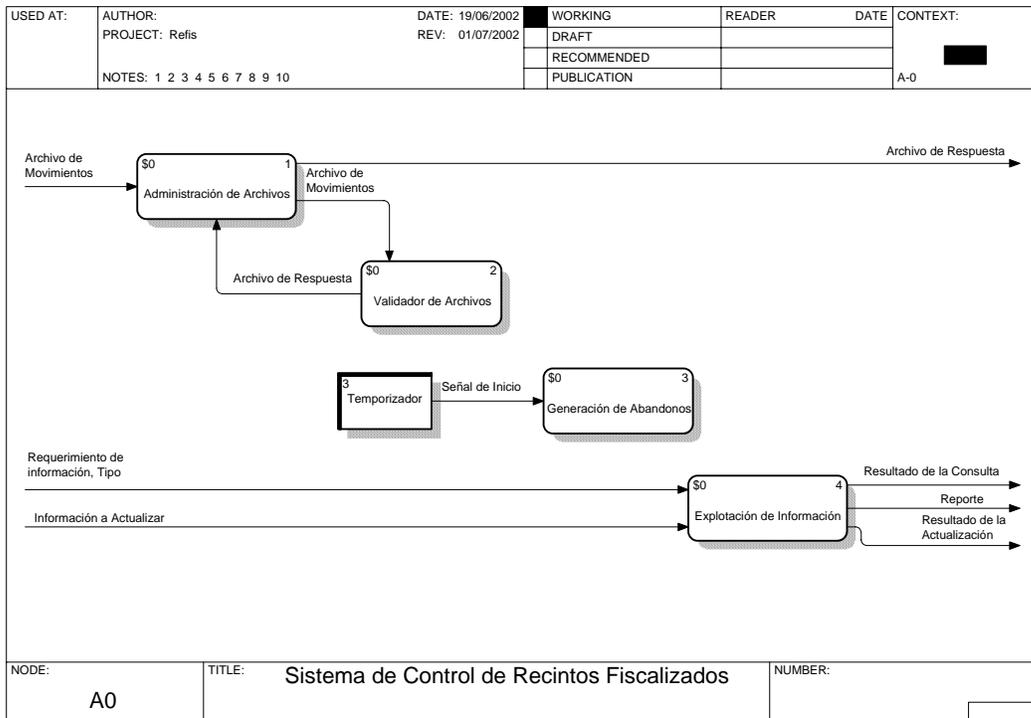


figura 2.2.20 DFD de Nivel 1 (Expansión del diagrama de contexto)

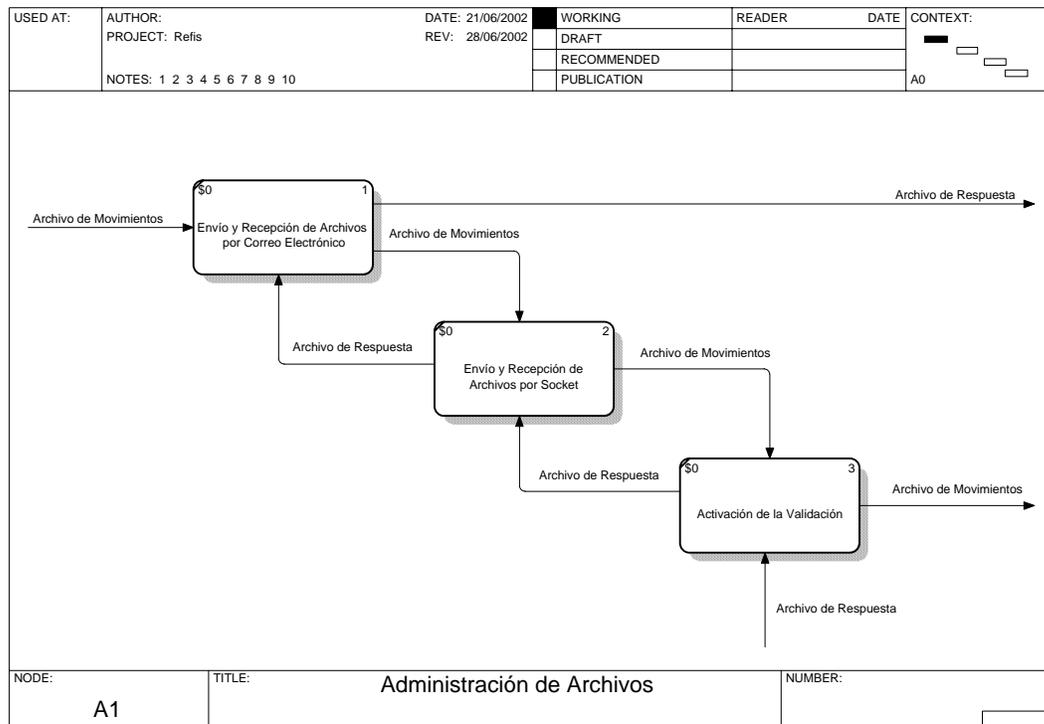


figura 2.2.21 DFD de Nivel 2, Administrador de Archivos.

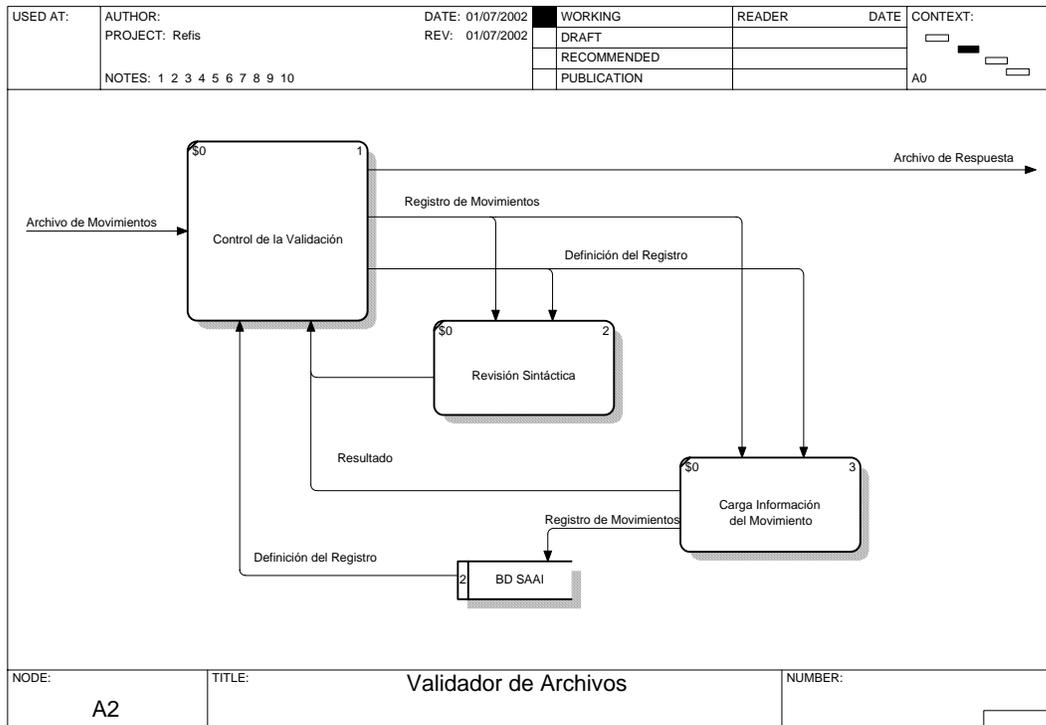


figura 2.2.22 DFD Nivel 2, Validador de Archivos

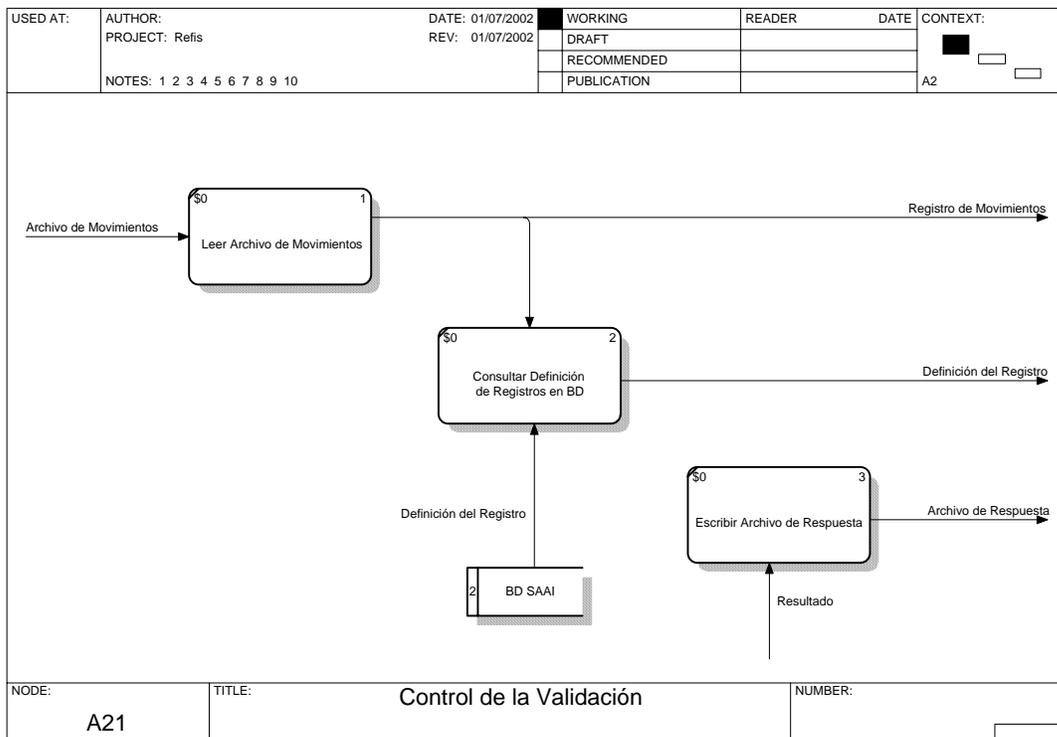


figura 2.2.23 DFD Nivel 3, Control de la Validación

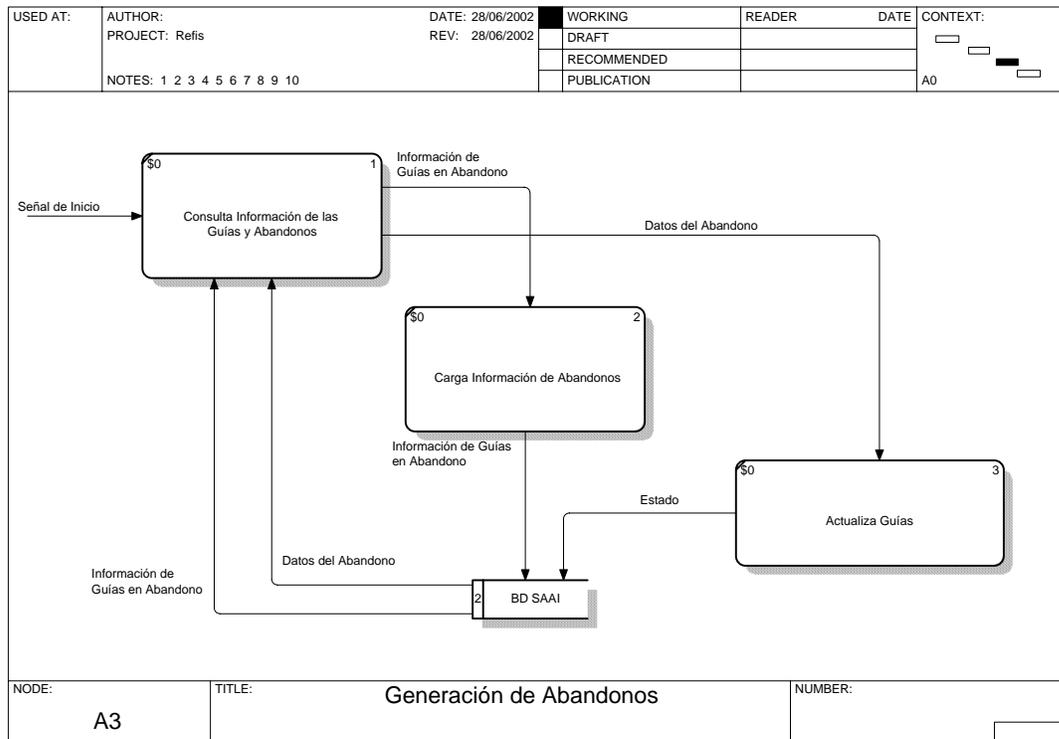


figura 2.2.24 DFD de Nivel 2, Generación de Abandonos

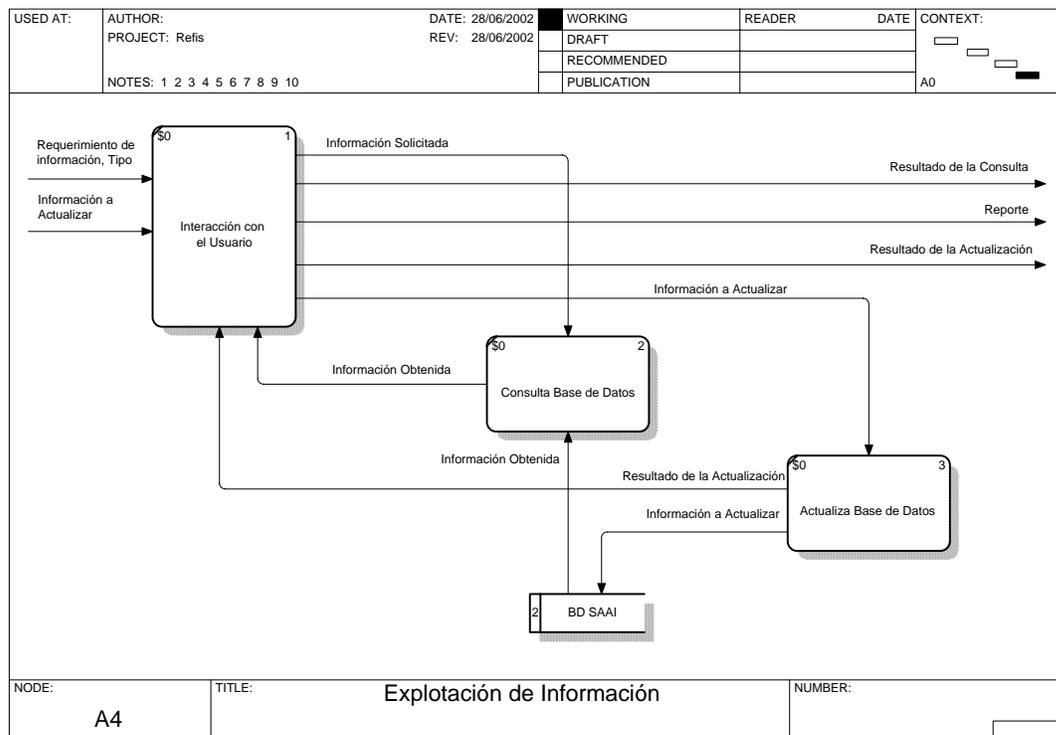


figura 2.2.25 DFD de Nivel 2, Explotación de Información.

2.3 Modelo de Comportamiento

Los modelos anteriores nos han descrito los datos y los procesos que los van a transformar, pero también es necesario identificar cómo responde el sistema a los estímulos del mundo exterior. La mayoría de los sistemas responden a situaciones externas o eventos que lo hacen cambiar de un estado, comportamiento observable desde el exterior, a otro (Dominio de Comportamiento).

El Diagrama de Transición de Estados (DTE) es el gráfico que permite representar cómo cambia el sistema de un estado a otro.

En las figura 2.3.1 a la 2.3.4 se muestran los diagramas de transición de estados para los cuatro procesos principales.

Diagramas de transición de estados

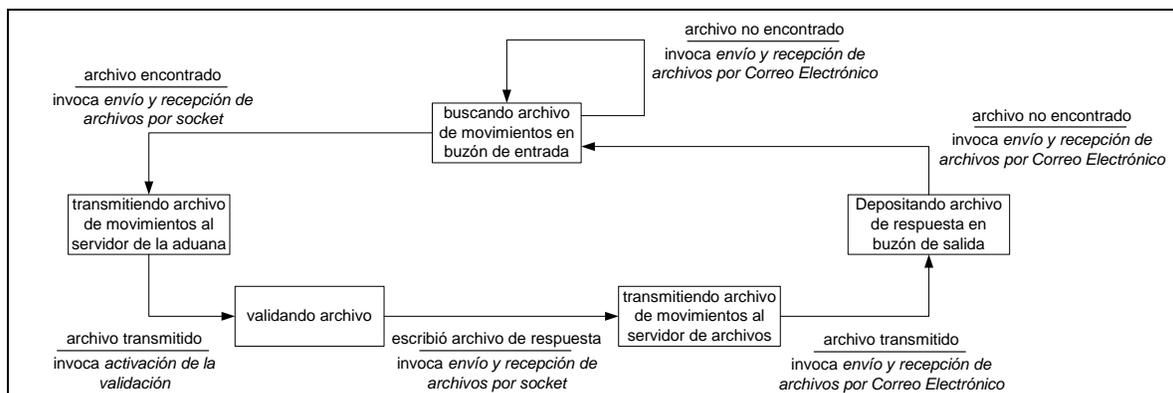


figura 2.3.1 DTE del Administración de Archivos

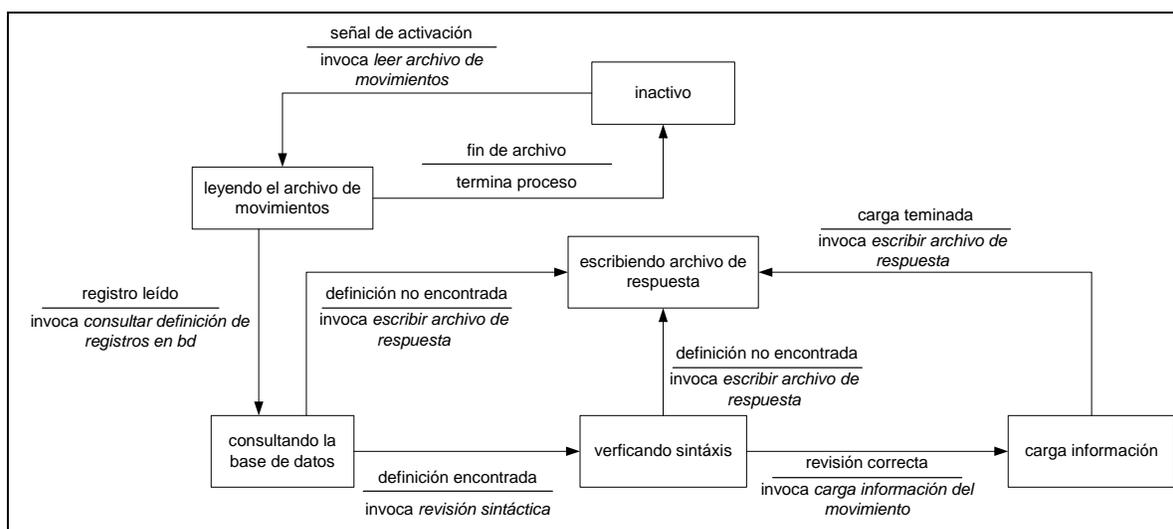


figura 2.3.2 DTE Validador de Archivos

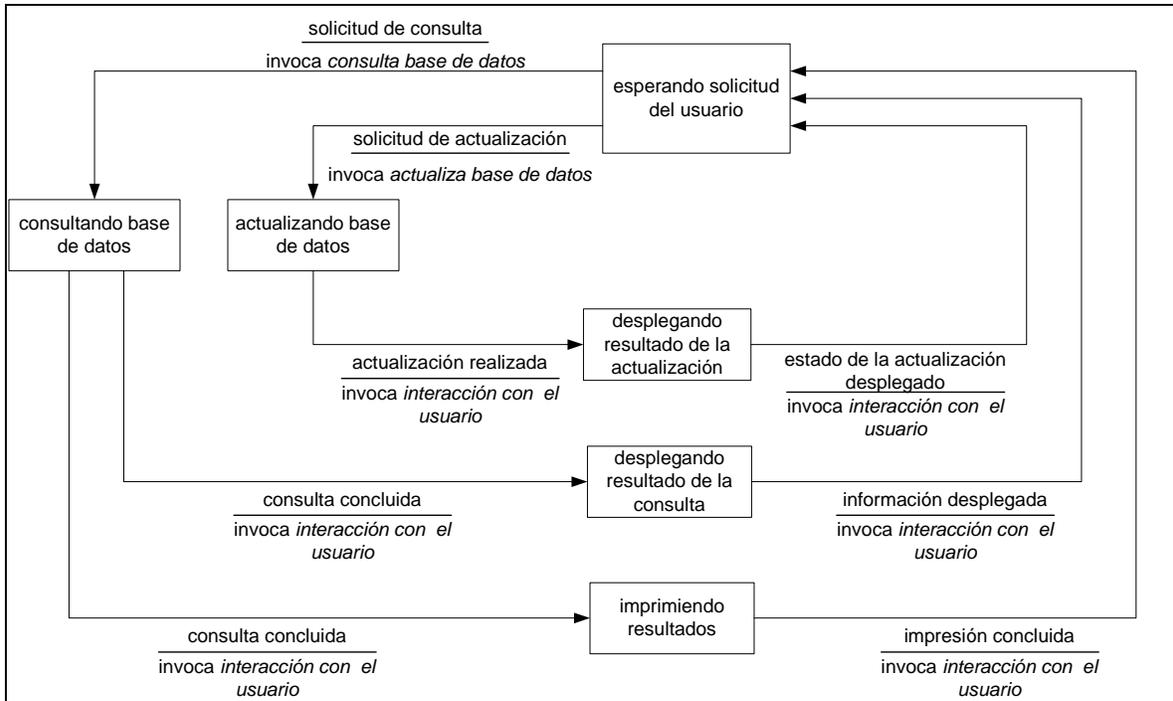


figura 2.3.3 DTE de la Explotación de Información

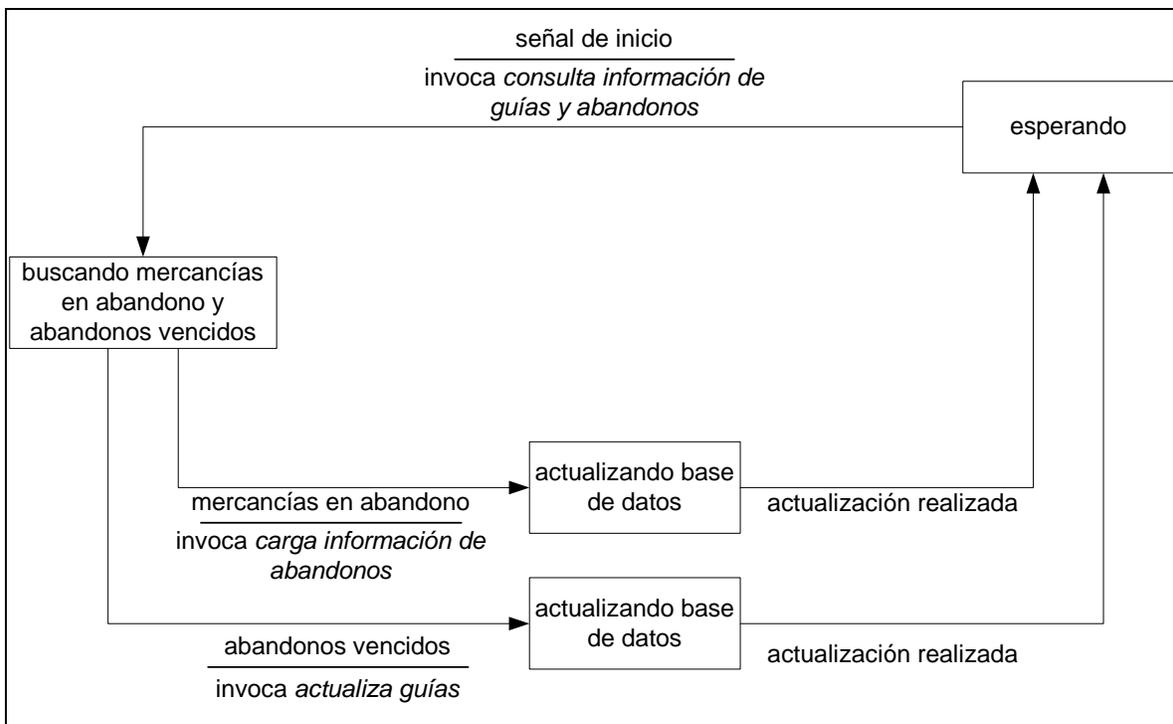


figura 2.3.4 DTE de la Generación de Abandonos

2.4 Prototipo

Aparte de los modelos descritos en los puntos anteriores, en ocasiones es necesario saber si el análisis hecho de los requerimientos se ajusta a las necesidades reales del usuario, es decir, si resuelven sus necesidades y si operativamente es factible, ya que se pueden cubrir estas necesidades pero entorpecer la operación del área del negocio que se está atendiendo. Es por ello que en ocasiones es recomendable desarrollar un modelo del sistema a construirse con el cual se pueda demostrar al usuario como se van a atender los requerimientos, así el usuario tendrá un acercamiento con el sistema a fin de que retroalimente, indicando si se cumplen o no de la manera esperada, para que éstos sean considerados en el refinamiento del sistema

Las ventajas que se obtiene al emplear prototipos son las siguientes:

- El usuario tiene un primer acercamiento con el sistema desde la primera fase del ciclo de vida.
- Identificar con oportunidad que el sistema desarrollado no sirve o no cumple los requerimientos, para desarrollar otro que se ajuste mejor a las necesidades.
- Las modificaciones al sistema durante su desarrollo se harán más rápidamente.

Es importante considerar que no para todos los sistemas es factible o en ocasiones no es conveniente la construcción de prototipos, esto dependerá del área de aplicación, la complejidad, de las características del usuario o cliente, etcétera. En general, cualquier aplicación que despliega información o que interactúa de manera considerable con el usuario, puede ser considerado como un candidato para la construcción de un prototipo.

Para que un prototipo sea efectivo, éste debe desarrollarse rápidamente para que el cliente o usuario puedan evaluar los resultados obtenidos y emitir los cambios necesarios. Por otro lado, debe existir un verdadero compromiso por parte del cliente para evaluar en tiempo el prototipo presentado.

En caso del problema que motiva el presente trabajo, se optó por construir el prototipo sólo del cliente para el módulo de Explotación de Información (ver figura 2.3.3) ya que los demás módulos operan de manera automatizada y no interactúan con el usuario. De hecho, con este módulo se podrá visualizar y evaluar la operación de los demás módulos

Para no hacer tan extenso el presente trabajo se muestran las partes más relevantes del prototipo desarrollado (figuras 2.4.1 a la 2.4.8) :

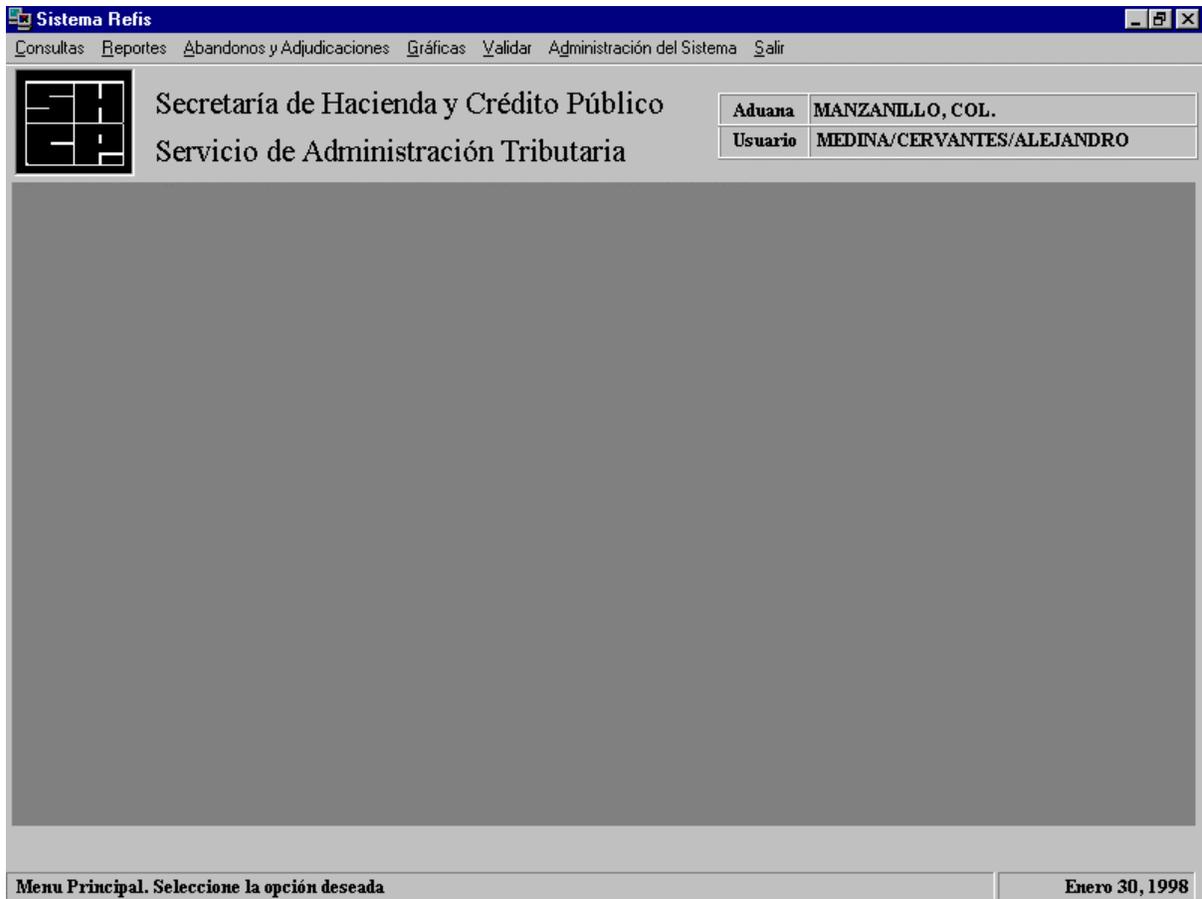


figura 2.4.1 Pantalla Principal



figura 2.4.2 Opciones del menú de consulta

Aprovechamientos

REFI: 01160 ALMACEN UNO Año: 1997

Buscar Salir

	Fecha		Ingresos	Monto		
	Mes	Año	Brutos	Compensable x Inv.	Compensable x Alm.	Pagado x Aprov.
▶	01	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$1,000.00
	02	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$2,000.00
	03	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$3,000.00
	04	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$4,000.00
	05	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$5,000.00
	06	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$6,000.00
	07	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$7,000.00
	08	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$8,000.00
	09	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$9,000.00
	10	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$10,000.00
	11	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$11,000.00
	12	1997	\$50,000.00	\$1,000.00	\$1,000.00	\$12,000.00

figura 2.4.5 Consulta de Aprovechamientos

Reportes

- Entradas
- Notificación de Arribo
- Salidas
- Mercancías ▶
- Aprovechamientos

figura 2.4.6 Opciones del Menú de Reportes

Período (Entradas)

REFI: * TODOS Período: Fecha Inicio: 06/02/1998 Fecha Fin: 06/02/1998

Tipo de Entrada: * - TODOS

Campos del reporte:

- Aduana orig.:
- Agente aduanal:
- Cantidad:
- Costo almacenaje:
- Destinatario:
- Fec. entrada orig.:
- Fecha entrada:
- Folio acta/PAMA:
- Folio irregularidad:
- Guía House:

Campos seleccionados:

- Aduana orig.:
- Agente aduanal:
- Cantidad:
- Fecha entrada:
- Guía House:

Impresora Aceptar Cancelar

Número de copias: 1

figura 2.4.7 Reporte de Entradas

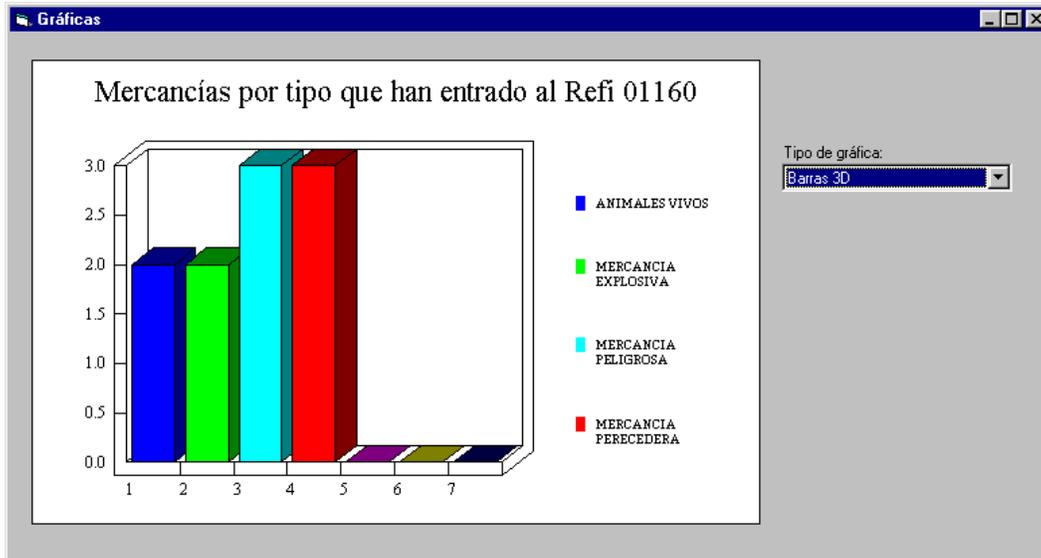


figura 2.4.8 Gráficas

Capítulo 3. Diseño

Capítulo 3. Diseño

El objetivo de la fase de diseño es obtener una representación o modelo de un sistema que será construido posteriormente.

El diseño de software es un proceso a través del cual los modelos obtenidos de la fase de análisis son transformados para obtener los “planos” para la construcción del sistema (arquitectura del sistema). Cada uno de los elementos del modelo de análisis proporcionan la información necesaria para obtener el modelo de diseño.

Este proceso es iterativo, ya que originalmente se cuenta con un modelo a nivel de abstracción alto o nivel de detalle muy general, conforme se va avanzado en el diseño, el refinamiento subsiguiente realizado a los modelos generará representaciones a un nivel de detalle más bajo o específico.

De la fase de diseño obtendremos la representación de la Estructura de Datos a ser empleada, la Estructura del Sistema, las características de las interfaces y la especificación de los procedimientos o algoritmos a ser codificados en la fase de construcción. Así mismo, es necesario establecer los indicadores que permitan evaluar los aspectos de funcionamiento del sistema y cómo se llevarán a cabo las pruebas para ello.

3.1 Arquitectura del Sistema

Como se mencionó, la fase de diseño proporcionará los “planos” para la construcción del sistema, los cuales conforman la arquitectura del sistema, que no es más que la representación jerárquica de los componentes (módulos) de la aplicación, la manera de interactuar entre ellos y las estructuras de datos empleados por éstos.

3.1.1 Estructura de Datos

Las estructuras se obtienen del modelo de datos elaborado en la fase de análisis, es decir, se emplean las entidades y relaciones definidas en el diagrama Entidad-Relación y el contenido del diccionario de datos.

Otro parámetro que determinará las estructuras de datos a definirse es la arquitectura de software y hardware donde se implementará la aplicación, normalmente dependerá del manejador de base de datos que se utilizará.

El sistema a desarrollarse en el presente trabajo, empleará como manejador de base de datos el de INFORMIX para HP-UX (UNIX de Hewlett Packard), el cual cuenta con los siguientes tipos de datos (tabla 3.1.1):

Tipos de Datos*

SMALLINT	Permite representar números enteros comprendidos en el rango de -32,767 hasta 37,767.
INTEGER	Permite representar números enteros comprendidos en el rango de -2,147,483,648 hasta 2,147,483,647.
SERIAL	Este tipo de datos es un tipo INTEGER.
FLOAT	Permite representar números reales de punto flotante, en caso de UNIX emplea 8 bytes para representarlos y que van desde 1.17549435 E-38 hasta 13.40282347 E+38.
DECIMAL(p,d)	Este tipo de datos permite representar números de punto fijo, donde p es el total de números a ser almacenados (la precisión), máximo 32 dígitos, y d el número de decimales deseados.
MONEY(p,d)	Es idéntico al tipo DECIMAL solo que el manejador transforma este tipo de datos a carácter al momento de desplegarlo y le añade el signo de moneda (-\$-por defecto).
CHAR(n)	Con este tipo de datos se pueden definir cadenas de caracteres de longitud n , donde $1 \leq n \leq 32,767$.
DATE	Este tipo de dato almacena una fecha como un entero (INTEGER), la cual puede ser formateada al momento de desplegarse.

tabla 3.1.1 Tipos de datos empleados por INFORMIX

* Solo se muestran los tipos de datos empleados para el sistema.

Empleando los tipos de datos disponibles y la información obtenida del modelo de datos, se pueden diseñar las tablas y catálogos que serán utilizadas por el sistema (tabla 3.1.2 a la 3.1.16).

Tablas del Sistema

Tabla: acceso_oper

Descripción: Recursos del sistema a los que tiene acceso el operador

Columna	Tipo	Descripción
aco_ope_cve	char(13)	Clave del Operador
aco_opcion	char(13)	Recurso al que se tiene acceso

tabla 3.1.2 Catálogo de recursos accesibles por operador

Tabla: aduanas

Descripción: Catálogo de Aduanas

Columna	Tipo	Descripción
ad_cve	char(3)	Clave de la Aduana.
ad_desc	char(80)	Nombre de la aduana.

tabla 3.1.3 Catálogo de aduanas

Tabla: aprovechamiento**Descripción:** Montos de aprovechamiento mensual declarados por cada Recinto Fiscalizado.

Columna	Tipo	Descripción
apr_car_cve	char(2)	Clave del Recinto Fiscalizado
apr_ad_cve	char(3)	Clave de la Aduana.
apr_mes	char(2)	Mes que se declara.
apr_anio	char(4)	Año que se declara.
apr_ingreso_bruto	integer	Ingreso bruto durante el mes declarado.
apr_monto_inv	integer	Monto compensable por inversión en el mes declarado.
apr_monto_alm	integer	Monto compensable por almacenaje en el mes declarado.
apr_monto_pagado	integer	Importe del aprovechamiento en el mes declarado.

*tabla 3.1.4 Tabla de aprovechamientos***Tabla:** cat_entradas**Descripción:** Catálogo del tipo de entradas

Columna	Tipo	Descripción
cae_cve	char(1)	Clave del tipo de entrada
cae_desc	char(45)	Descripción del tipo de entrada

*tabla 3.1.5 Catálogo de tipos de entradas***Tabla:** cat_mercancias**Descripción:** Catálogo del Tipo de Mercancías

Columna	Tipo	Descripción
cam_cve	char(1)	Clave del tipo de mercancía.
cam_desc	char(30)	Descripción del tipo de mercancía.
cam_plazo	integer	Plazo en días para retirar la mercancía del Recinto Fiscalizado.
cam_peligrosa	char(1)	Bandera de mercancía peligrosa 0 = No Peligrosa. 1 = Peligrosa.

*tabla 3.1.6 Catálogo de tipos de mercancía***Tabla:** cat_refis**Descripción:** Catálogo de Recintos Fiscalizados

Columna	Tipo	Descripción
car_llave	char(8)	Llave
car_email_addr	char(120)	Dirección de correo electrónico.
car_fecalta	date	Fecha en que se dio de alta el Recinto Fiscalizado.

car_fecbaja	date	Fecha en que se dio de baja el Recinto Fiscalizado.
car_cve	char(2)	Clave del Recinto Fiscalizado
car_domicilio	char(80)	Domicilio del Recinto Fiscalizado.
car_ad_cve	char(3)	Clave de la Aduana.
car_nom_refi	char(40)	Nombre del Recinto Fiscalizado.

tabla 3.1.7 Catálogo de Recintos Fiscalizados

Tabla: cat_salidas

Descripción: Catálogo del Tipo de Salidas

Columna	Tipo	Descripción
cas_cve	char(1)	Clave del tipo de salida o extracción.
cas_desc	char(45)	Descripción del tipo de salida o extracción.

tabla 3.1.8 Catálogo de tipos de extracción

Tabla: guia_abandono_adj

Descripción: Guías en abandono

Columna	Tipo	Descripción
gaa_car_cve	char(2)	Clave del Recinto Fiscalizado
gaa_ad_cve	char(3)	Clave de la Aduana.
gaa_gui_reg_ent	char(12)	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
gaa_gui_fec_ent	date	Fecha en que ingresó la guía al Recinto Fiscalizado
gaa_fec_ab	date	Fecha en que la mercancía causó abandono.
gaa_ope_cve	char(13)	Operador que generó la notificación de abandono.
gaa_fec_gnotif_ab	date	Fecha en que se generó la notificación de abandono.
gaa_folio_ab	integer	Folio de la notificación.
gaa_notif_ab	char(12)	Número de gafete del Notificador
gaa_fec_notif_ab	date	Fecha en que se notificó del abandono al Destinatario/Consignatario.
gaa_fec_cont_ab	date	Fecha en la que el Destinatario/Consignatario promovió una controversia.
gaa_fec_resol_ab	date	Fecha en que se dictó la resolución a la controversia promovida por el Destinatario/Consignatario.
gaa_inst_ab	char(80)	Instancia ante la que se promovió la controversia.

gaa_folio_res_ab	char(25)	Folio asignado por la instancia a la resolución emitida por ella.
gaa_status	integer	Estado en el que se encuentra el abandono.

tabla 3.1.9 Tabla de guías en abandono y/o adjudicadas

Tabla: guia_notificacion

Descripción Control de las notificaciones de arribo hechas a los destinatarios/consignatarios de las mercancías.

Columna	Tipo	Descripción
gun_car_cve	char(2)	Clave del Recinto Fiscalizado
gun_ad_cve	char(3)	Clave de la Aduana.
gun_gui_reg_ent	char(12)	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
gun_gui_fec_ent	date	Fecha en que ingresó la guía al Recinto Fiscalizado
gun_fec_notif	date	Fecha en que se notificó al destinatario/consignatario que las mercancías arribaron al Recinto Fiscalizado.
gun_tipo_aviso	char(1)	Cómo se notificó al destinatario/consignatario que las mercancías arribaron al Recinto Fiscalizado.
gun_calle_dest	char(40)	Calle donde se ubica el domicilio del destinatario/consignatario de las mercancías.
gun_col_dest	char(30)	Colonia donde se ubica el domicilio del destinatario/consignatario de las mercancías.
gun_cp_dest	char(5)	Código Postal donde se ubica el domicilio del destinatario/consignatario de las mercancías.
gun_cd_dest	char(20)	Ciudad donde se ubica el domicilio del destinatario/consignatario de las mercancías.
gun_edo_dest	char(10)	Estado donde se ubica el domicilio del destinatario/consignatario de las mercancías.
gun_pa_cve	char(3)	Clave del País

tabla 3.1.10 Tabla de guías con notificación de arribo

Tabla: guia_salida**Descripción:** Guías retiradas del Recinto Fiscalizado.

Columna	Tipo	Descripción
gus_agente	char(4)	Agente Aduanal que tramitó el pedimento con el cual se extrae la mercancía.
gus_numdcto	char(7)	Número de pedimento con el cual se extrae la mercancía.
gus_aduana_dest	char(3)	Aduana en la que se ubica el Recinto Fiscalizado al que se dirige la mercancía en caso de una transferencia.
gus_refi_dest	char(2)	Recinto Fiscalizado al que se dirige la mercancía en caso de una transferencia.
gus_cantidad	decimal(11,3)	Cantidad de mercancía extraída en unidades de medida
gus_folio_acta	char(12)	Folio del Acta con la que se extrae la mercancía.
gus_um_cve	char(2)	Clave de la unidad de medida.
gus_cant_merma	decimal(11,3)	Cantidad de mercancía faltante en unidades de medida
gus_ad_cve	char(3)	Clave de la Aduana.
gus_gui_reg_ent	char(12)	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
gus_car_cve	char(2)	Clave del Recinto Fiscalizado
gus_fec_sal	date	Fecha en que extrae la mercancía del Recinto Fiscalizado.
gus_cas_cve	char(1)	Clave del tipo de salida o extracción.
gus_gui_fec_ent	date	Fecha en que ingresó la guía al Recinto Fiscalizado
gus_ultima	char(1)	Bandera que indica si la subdivisión extraída es la última.
gus_num_sub	integer	Número de subdivisión extraída.

tabla 3.1.11 Tabla de guías retiradas del Recinto Fiscalizado**Tabla:** guias**Descripción:** Guías que ingresan a los Recintos Fiscalizados

Columna	Tipo	Descripción
gui_car_cve	char(2)	Clave del Recinto Fiscalizado
gui_ad_cve	char(3)	Clave de la Aduana.
gui_reg_ent	char(12)	Registro de Entrada asignado a la guía por el Recinto Fiscalizado.
gui_fec_ent	date	Fecha en que ingresó la guía al

Columna	Tipo	Descripción
		Recinto Fiscalizado
gui_cae_cve	char(1)	Clave del tipo de entrada
gui_tipo		Clave del tipo de guía
gui_cam_cve		Clave del tipo de mercancía.
gui_piezas	integer	Total de piezas contenidas en la guía.
gui_um_cve	char(2)	Clave de la unidad de medida.
gui_cantidad	decimal(11,3)	Cantidad de mercancía en unidades de medida
gui_master	char(16)	Número de la guía master.
gui_house	char(35)	Número de la guía house.
gui_folio_acta	char(17)	Folio del Acta con la ingresa la mercancía al Recinto Fiscalizado.
gui_costo_alm	integer	Costo diario promedio de almacenaje que causa la mercancía.
gui_refi_orig	char(2)	Clave del Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
gui_aduana_orig	char(3)	Clave de la aduana donde se ubica el Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
gui_reg_ent_orig	char(12)	Número de Registro asignado por Recinto Fiscalizado de donde proviene la mercancía en caso de ingreso por transferencia.
gui_fec_ent_orig	date	Fecha en que ingreso la mercancía al país en caso de ingreso por transferencia.
gui_agente	char(4)	Patente del Agente Aduanal que tramitó el pedimento con el cual ingresó la mercancía al Recinto Fiscalizado.
gui_numdcto	char(7)	Número de pedimento con el cual ingresó la mercancía al Recinto Fiscalizado.
gui_rfe_rfc	char(13)	Registro Federal de Causante del Destinatario/Consignatario de las mercancías.
gui_nom_dest	char(40)	Nombre o Razón Social del Destinatario/Consignatario de las mercancías.
gui_tipo_dom	char(1)	Tipo de domicilio del Destinatario/Consignatario de las

Columna	Tipo	Descripción
		mercancías.
gui_folio_irr	char(6)	Folio del Acta en la cual se registraron las irregularidades encontradas.
gui_status	integer	Estado en el que se encuentra la guía.

tabla 3.1.12 Tabla de guías

Tabla: operadores

Descripción: Catálogo de los operadores del sistema.

Columna	Tipo	Descripción
ope_cve	char(13)	Clave del Operador
ope_pue_cve	char(2)	Clave del puesto.
ope_gafete	char(17)	Número de gafete asignado al operador.
ope_rfc	char(13)	Registro Federal de Causantes del operador.
ope_nom_oper	char(40)	Nombre del operador.
ope_passwd	char(13)	Contraseña de acceso al sistema.

tabla 3.1.13 Catálogo de operadores

Tabla: países

Descripción: Catálogo de países.

Columna	Tipo	Descripción
pa_cve	char(3)	Clave del País
pa_desc	char(80)	Nombre del País

tabla 3.1.14 Catálogo de países

Tabla: puestos

Descripción: Catálogo de puestos.

Columna	Tipo	Descripción
pue_cve	char(2)	Clave del puesto.
pue_desc	char(30)	Descripción del puesto.

tabla 3.1.15 Catálogo de puestos

Tabla: unid_medida

Descripción: Catálogo de unidades de medida.

Columna	Tipo	Descripción
um_cve	char(2)	Clave de la unidad de medida.
um_desc	char(80)	Descripción de la unidad de medida.

tabla 3.1.16 Catálogo de unidades de medida

3.1.2 Estructuras del Sistema

Es una representación de cómo se encuentran organizados los componentes o módulos del sistema, uno de los modelos empleados para esta representación es el de árbol, denominado carta estructurada.

Cartas Estructuradas

Las cartas estructuradas es una notación que sirve para representar la jerarquía de control del sistema. Sólo muestran la organización de los componentes del sistema (módulos), pero no la secuencia de los procesos, orden de decisiones o repetición de operaciones.

A continuación se muestran las cartas estructuradas del “Sistema de Control de Recintos Fiscalizados” (figura 3.1.1 a la figura 3.1.6).

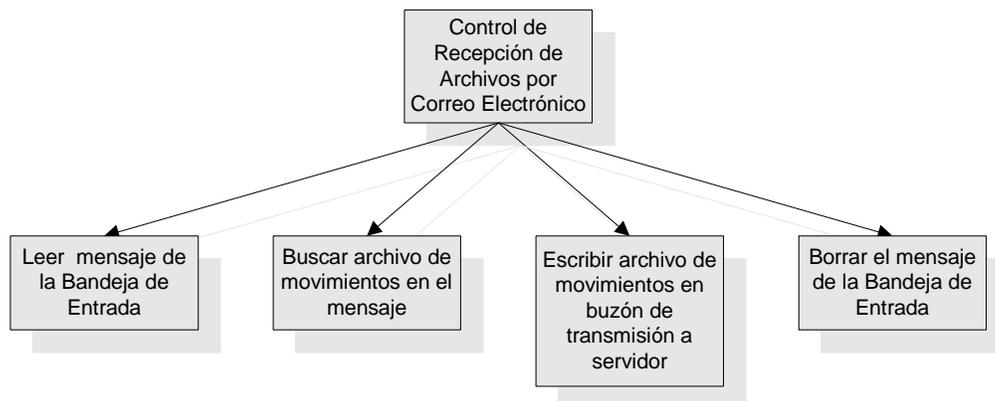


figura 3.1.1 Carta Estructurada Control de Recepción de Archivos por Correo Electrónico

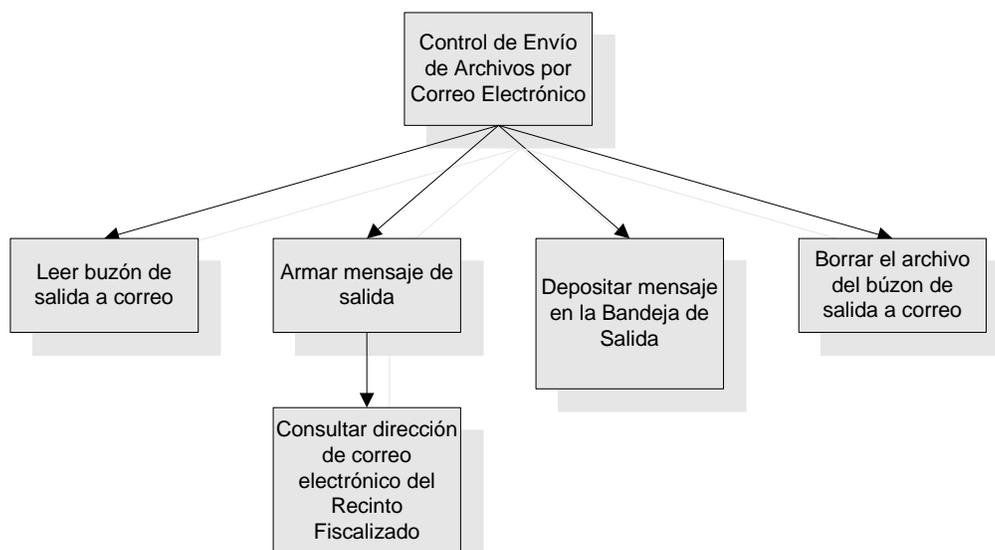


figura 3.1.2 Carta Estructurada Control de Envío de Archivos por Correo Electrónico

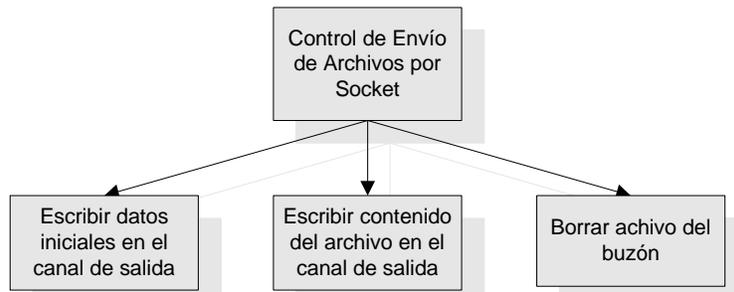


figura 3.1.3 Carta Estructurada Control de Envío de Archivos por Socket

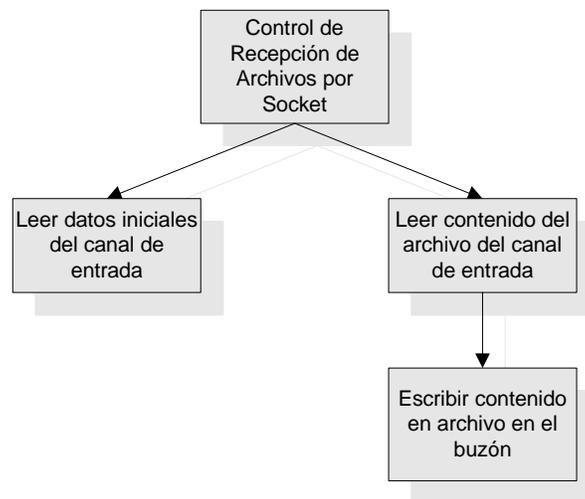


figura 3.1.4 Carta Estructurada Control de Recepción de Archivos por Socket

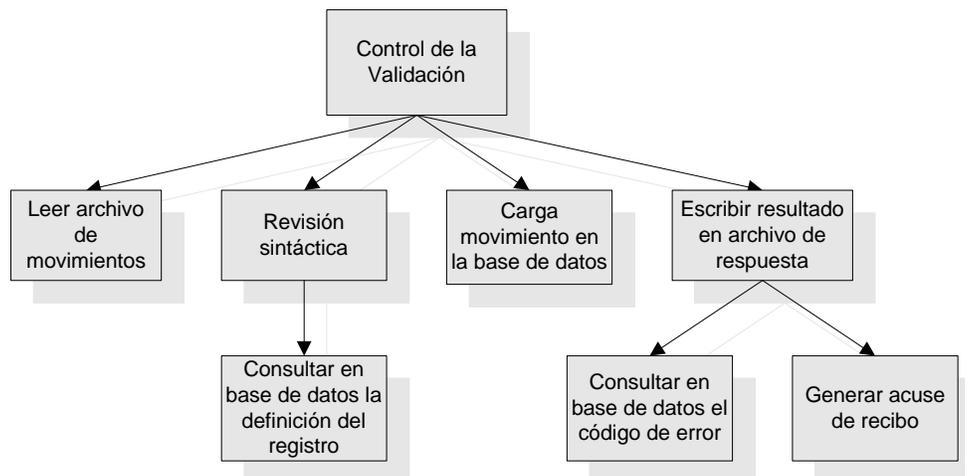


figura 3.1.5 Carta Estructurada del Control de la Validación

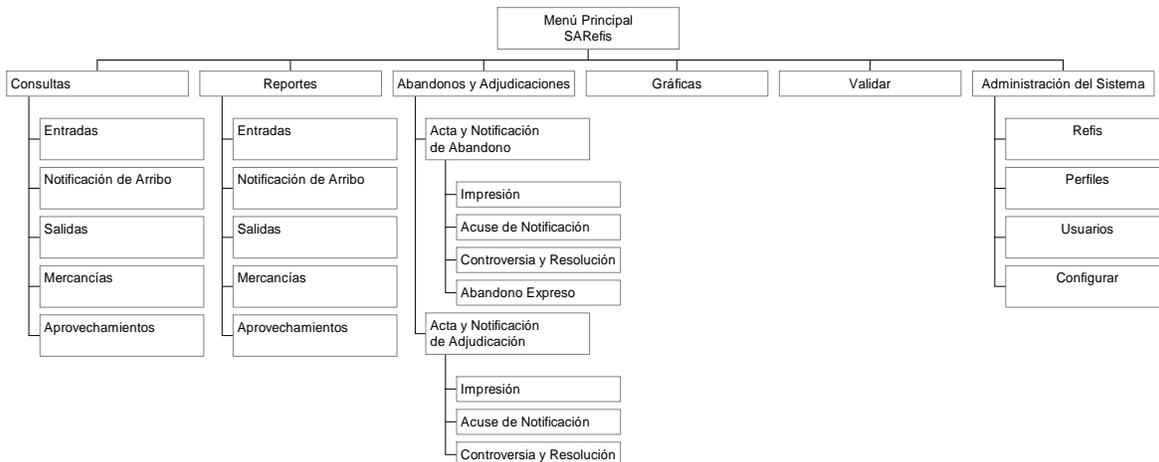


figura 3.1.6 Estructura del Cliente para Explotación de Información

3.2 Diseño de las Interfaces del Sistema

El diseño de las interfaces del sistema tiene como finalidad describir como interactúan los módulos del sistema entre sí (interfaces internas), como interactúa el sistema con las entidades externas que pueden ser otros sistemas (interfaces externas) o los usuarios propios del software a desarrollarse (interfaces hombre-máquina).

3.2.1 Diseño de las interfaces internas y externas

El diseño de las interfaces internas está definido por la manera en que los datos deben fluir entre los módulos. Durante el análisis se definió gran parte de la información requerida para el diseño de las interfaces tanto internas como externas. Así como las burbujas dentro de los DFDs representan las estructuras o módulos del sistema, las flechas que entran y salen de los procesos, son traducidas en las interfaces correspondientes para cada módulo.

En el caso del desarrollo estructurado, para la definición de las interfaces internas se utiliza la lista de argumentos como medio para transferir los datos entre los módulos.

Considerando que la restricción establecida para el acceso a la base de datos es no emplear ODBC, será necesario diseñar el cliente para ejecutar los comandos SQL y el servidor que las atienda, comunicándose entre ellos intercambiando mensajes a través de sockets Berkeley por TCP/IP.

DEFINICIÓN DE MENSAJES PARA EJECUCIÓN DE COMANDOS SQL

Operaciones de Base de Datos

Petición

Tamaño Mensaje	Código Operación	Buffer
4 Bytes	9 Bytes	Tamaño Mensaje – 8 Bytes

Apertura de Base de Datos		
9 + LEN(nombre Base Datos)	QRY_OPENDB "ATMTA1111"	Nombre de la BD
Control de Transacciones y Cierre de la BD		
10	QRY_DATOS "ATMTA0110"	SQLCLOSE 4 SQLBEGINTRANS 5 SQLCOMMIT 6 SQLROLLBACK 7

Ejecución de Sentencias INSERT, UPDATE y DELETE		
9 + LEN(Sentencia a ejecutar)	QRY_DATOS "ATMTA0110"	Sentencia a ejecutar

Respuesta

Tamaño Mensaje	Código Operación	Código de Retorno	Buffer
4 Bytes	9 Bytes	6 Bytes	Tamaño Buffer – 15 Bytes

15 + LEN(Descripción de Error)	QRY_CODIGO "ATMTA1001"	SQLCODE	Descripción del Error (Si SQLCODE <> 0)
--------------------------------	------------------------	---------	---

Operaciones de Consulta

Ejecución de Sentencias SELECT

Tamaño Mensaje	Buffer
4 Bytes	Tamaño Mensaje Bytes
4 Bytes + LEN(Sentencia SELECT)	Sentencia SELECT

Recibir Resultado de la Sentencia SELECT en un archivo

Tamaño Mensaje	Código Operación	Buffer
4 Bytes	9 Bytes	Tamaño Mensaje – 9 Bytes
9 + LEN(Sentencia SELECT)	QRY_ARCHIVO "ATMTA1000"	Sentencia SELECT

Desplazamiento a lo largo del CURSOR y Cerrar el CURSOR

Dirección
4 Bytes
SQLPRIMERO 0
SQLULTIMO 1
SQLSIGUIENTE 2
SQLANTERIOR 3
SQLCLOSE 4

Búsquedas a lo largo del CURSOR

Dirección de Búsqueda	Tamaño del Buffer	Buffer
4 Bytes	4 Bytes	Tamaño Condición
SQLFINDFIRST 9	LEN(Condición)	Condición de búsqueda
SQLFINDNEXT 10		
SQLFINDLAST 11		

Resultado

Tamaño Mensaje	Buffer
4 Bytes	Tamaño Mensaje Bytes
LEN(Resultado)	Resultado: Nombre de las columnas + QRY_EOD + Tipo de dato de las columnas + QRY_EOD + Longitud de las

	<p>columnas + QRY_EOD + Valores + No de Renglones</p> <p>Donde</p> <p>Nombre de las columnas = nombre_columna + " " [+ (nombre_columna + " ")*]</p> <p>Tipo de dato de las columnas = tipo_dato_columna + " " [+ (tipo_dato_columna + " ")*]</p> <p>tipo_dato_columna = "SQLCHAR" "SQLSMINT" "SQLINT" "SQLFLOAT" "SQLSMFLOAT" "SQLDECIMAL" "SQLSERIAL" "SQLDATE" "SQLMONEY" "SQLDTIME" "SQLBYTES" "SQLTEXT" "SQLVCHAR" "SQLINTERVAL" "SQLNCHAR" "SQLNVCHAR"</p> <p>Longitud de las columnas = digito_6 + " " [+ (digito_6 + " ")*]</p> <p>digito_6 = "0-9" + "0-9" + "0-9" + "0-9" + "0-9" + "0-9"</p> <p>No de Renglones = digito_6</p> <p>QRY_EOD = "ATMTA0000"</p>
--	--

En cuanto a las interfaces externas, el diseño comienza con la evaluación de cada una de las entidades externas representadas en los DFDs, los datos y controles necesarios para las entidades externas determinan las interfaces a diseñarse.

Para el sistema propuesto y remontándonos al Diagrama de Contexto representado en la **¡Error! No se encuentra el origen de la referencia.**, se distinguen dos entidades externas, Recinto Fiscalizado y Operador.

El Recinto Fiscalizado será la entidad encargada de "nutrir" o alimentar de información al sistema, indicando los movimientos de mercancías que pasan a través de ellos, los cuales serán entregados al sistema en archivos (archivo de movimientos) donde cada registro contenido en el archivo corresponderá a un

movimiento. De igual manera, el sistema entregará en archivo de respuesta con los acuses de los movimientos recibidos (aceptado o rechazado).

Lo anterior obliga a hacer una definición para los archivos intercambiados entre los Recintos Fiscalizados con el sistema a desarrollarse, así como de su contenido, enseguida se muestran las definiciones hechas al respecto:

Archivo de Movimientos

Cada archivo puede contener información del estatus de flujo de uno o más embarques, además de poderse transmitir durante el día un número ilimitado de estos archivos.

El nombre de cada archivo tiene la forma **asiinnnn.ddd**, donde:

- a Es una clave que indica que se trata de un archivo de REFI
- s Corresponde la sección aduanera a la que pertenezca el REFI
- ii Número del REFI asignado por la S.H.C.P.
- nnnn Número consecutivo del archivo. Se inicia a partir del 0001, en caso de se lleguen al archivo con números 9999, se deberá reiniciar la numeración a partir del 0001.
- ddd Es el día juliano en el que se efectúa la transmisión.
 - 001 corresponde al “1º de Enero”
 - 002 corresponde al “2 de Enero”
 - . . .
 - 365 corresponde al “31 de Diciembre” (Sí el año es bisiesto, sería 366).

El archivo es de texto, en formato ASCII y el separador de registros utilizado es el caracter Control-J (line feed, con código ASCII HEXADECIMAL=A, OCTAL=12, DECIMAL=10). No se aceptan caracteres acentuados y la letra “ñ” se sustituye por el caracter “#”.

REGISTROS DE DATOS DE ENTRADAS

TIPO “101”

ESTRUCTURA DEL REGISTRO DE DATOS DE ENTRADA

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
1	Tipo de Registro	3	Numérico	Sí	Identificación del tipo de registro
2	Clave del Refi	5	Numérico	Sí	Número de identificación del Refi
3	Registro de ingreso	12	Numérico	Sí	Clave de registro de ingreso de la mercancía (consecutivo)

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
4	Fecha de entrada	8	Numérico	Sí	Fecha en que se registra la entrada de la mercancía al territorio nacional
5	Fecha de ingreso	8	Numérico	Sí	Fecha en que se registra el ingreso de la mercancía al REFI
6	Tipo de Ingreso	1	Alfabético	Sí	Clave que identifica el tipo de ingreso (Según Catálogo)
7	Tipo de Operación	1	Numérico	Sí	Tipo de Operación (Importación o Exportación)
8	Guía Master	16	Alfanumérico	En caso de existir	Clave de la guía Master del embarque o del manifiesto de carga cuando se trate de aduanas marítimas
9	Guía House	35	Alfanumérico	En caso de existir	Clave de la guía House del embarque o del conocimiento de embarque
10	Tipo de mercancía	1	Alfabético	Sí	Clave que identifica el tipo de mercancía (Según Catálogo)
11	Piezas según guía	7	Numérico	Sí	Número de piezas especificadas en la guía
12	Unidad de medida según guía	2	Numérico	Sí	Unidad de medida utilizada en la guía (Según Catálogo)
13	Cantidad (unid. medida)	15	Numérico	Sí	Peso o cantidad según unidad de medida según guía
14	Folio acta de entrada o número de PAMA o número de oficio	17	Alfanumérico	En caso de existir	Puede emplearse este campo para poner el número de PAMA, Número de Oficio o Escrito
15	Recibo de Depósito	10	Alfanumérico	Dependiendo del Tipo de ingreso	Número de Recibo Oficial
16	Clave del Refi origen	5	Numérico	En caso de transferencias	Se emplea cuando se trate de una transferencia desde otro refi para identificar el refi destino
17	Registro de entrada origen	12	Numérico	En caso de transferencias	Se emplea cuando se trate de una transferencia desde otro refi para identificar el registro de entrada con que se registro en el refi origen
8	Agente aduanal	4	Numérico	En caso de entrada con pedimento	Número del agente aduanal si se registra con pedimento
19	Número de pedimento	7	Numérico	En caso de entrada con pedimento	Número de pedimento si se realiza con pedimento
20	Aduana-Sección origen	3	Numérico	En caso de entrada con pedimento de Tránsito	Se emplea cuando es un tránsito para identificar la aduana origen del tránsito (Miscelánea Fiscal de Comercio Exterior Anexo 22 Apéndice 4)
21	Nombre destinatario o consignatario	40	Alfanumérico	Sí	Nombre del destinatario o consignatario

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
22	Tipo de domicilio	1	Alfabético	Sí	A Domicilio Competo B Domicilio Parcial C Sin Domicilio
23	Clave de irregularidad	1	Alfabético	Sí	Clave de la irregularidad según catálogo
24	Firma de Autenticidad	15	Alfanumérico	A futuro	Firma de Autenticidad del Agente o Apoderado Aduanal

**REGISTROS DE DATOS DE NOTIFICACIONES DE
ARRIBO DE MERCANCÍAS**

TIPO "102"

ESTRUCTURA DEL REGISTRO DE DATOS DE NOTIFICACIÓN

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
1	Tipo de Registro	3	Numérico	Sí	Identificación del tipo de registro
2	Clave del Refi	5	Numérico	Sí	Número de identificación del Refi
3	Registro de ingreso	12	Numérico	Sí	Clave de registro de ingreso de la mercancía al REFI
4	Fecha de entrada	8	Numérico	Sí	Fecha en que se registra la entrada de la mercancía al territorio nacional
5	Fecha de notificación	8	Numérico	Sí	Fecha en que se hizo la notificación
6	Tipo de aviso	1	Alfabético	Sí	Tipo de aviso
7	Calle y número	40	Alfanumérico	No	Calle y número del destinatario o consignatario
8	Colonia	30	Alfanumérico	No	Colonia del destinatario o consignatario
9	Código postal	5	Numérico	No	Código postal del destinatario o consignatario
10	Ciudad	20	Alfabético	No	Ciudad del destinatario o consignatario
11	Entidad Federativa	10	Alfabético	No	Entidad Federativa en donde radica del destinatario o consignatario
12	Firma de Autenticidad	15	Alfanumérico	A futuro	Firma de Autenticidad del Agente o Apoderado Aduanal

REGISTROS DE DATOS DE SALIDAS

TIPO "103"

ESTRUCTURA DEL REGISTRO DE DATOS DE SALIDAS

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
1	Tipo de Registro	3	Numérico	Sí	Identificación del tipo de registro
2	Clave del Refi	5	Numérico	Sí	Número de identificación del Refi
3	Registro de ingreso	12	Numérico	Sí	Clave de registro de ingreso de la

	Nombre del campo	Long	Tipo	Obligatorio	Descripción
					mercancía
4	Fecha de entrada	8	Numérico	Sí	Fecha en que se registra la entrada de la mercancía al territorio nacional
5	Número de subdivisión	4	Numérico	Sí	Número de subdivisión que sale
6	Ultima parcialidad	1	Alfabético	Sí	Identifica si es la última subdivisión
7	Fecha de salida	8	Numérico	Sí	Fecha de salida de la mercancía
8	Tipo de salida	1	Alfabético	Sí	Clave del tipo de salida
9	Clave del Refi Destino	5	Numérico	En caso de transferencias	Número de identificación del Refi al que se transfieren las mercancías
10	Agente	4	Numérico	En caso de existir	Número del agente aduanal si él registra con pedimento
11	Número de documento	7	Numérico	En caso de existir	Número de pedimento si se realiza con pedimento
12	Folio acta de salida u oficio emitido por la autoridad	12	Alfanumérico	En caso de existir	Número de acta con la que sale la mercancía
13	Costo diario de almacenaje	7	Numérico	No	En caso de ser un PAMA o que la mercancía genere un cargo que posteriormente se acredite sobre el aprovechamiento, se introduce el costo de almacenaje diario
14	Unidad de medida guía	2	Numérico	Sí	Unidad de medida utilizada en la guía
15	Cantidad	11	Numérico	Sí	Peso o cantidad según unidad de medida
16	Cantidad de merma	11	Numérico	No	Peso o cantidad de merma, cuando se trate de aduanas marítimas o terrestres.
17	Firma de Autenticidad	15	Alfanumérico	A futuro	Firma de Autenticidad del Agente o Apoderado Aduanal

Archivo de Respuesta

Por cada archivo transmitido al equipo REFIS SHCP, se generará un archivo de respuesta con el mismo nombre, pero con la extensión “.err”.

Por lo tanto, el formato del nombre será : **asiinnnn.err**

Estructura del registro de contestación

REGISTROS DE RESPUESTA EMITIDO POR EL EQUIPO REFIS SHCP

Descripción	Longitud	Valor
Llave del registro	Variable	Depende de la llave definida
Tipo de registro	3	Para el caso de refis (101 a 104)

Descripción	Longitud	Valor
Tipo de respuesta	1	F = Firma electrónica, E = Código de error, C = registro de control
Valor de retorno	8	Si Tipo de respuesta = F, los ocho caracteres representan la firma electrónica Si Tipo de respuesta = E, los primeros cuatro caracteres corresponden al número de campo y los otros cuatro al código de error (ver catálogo de errores) Si Tipo de respuesta = C, los ocho caracteres representan el número de registros procesados, este tipo de registro es obligatorio al final de cada archivo de respuesta validado correctamente.
Fecha y hora de validación	14	Fecha y hora en que fue validado el registro los primeros ocho caracteres corresponden a la fecha y los demás corresponden a la hora, si el tipo de respuesta = C no se ocupan

Registro	Llave	Longitud
Entrada Notificación Rectificación	Clave del Refi Registro de Ingreso Fecha de entrada	25
Salida	Clave del Refi Registro de Ingreso Fecha de entrada No. de subdivisión	29
Aprovechamiento	Clave del Refi Mes de la declaración Año de la declaración	11
Control		0

Envío de E-Mail

- El mensaje debe contener en el remitente (sender) el nombre del origen. Los Recintos Fiscalizados pondrán su razón social, el sistema incluirá la leyenda "Sistema Automatizado de Recintos Fiscalizados".
- En el asunto (subject) se especificará el tipo de archivo que se transmite y nombre del mismo. Los recintos fiscalizados mandarían la leyenda "Archivo de SARefis: nombre_archivo", por su parte el sistema pondrá "Archivo de Respuesta SARefis: nombre_archivo".
- Se incluirá solo un archivo por mensaje a transmitirse (attachment), para ello, se empleará el estándar de codificación MIME.
- El protocolo empleado para enviar el mensaje será SMTP (Simple Mail Transfer Protocol).

Recepción de E-Mail

- Se extraerá del buzón el mensaje recibido empleando el protocolo POP3 (Post Office Protocol Versión 3).

Para decodificar el archivo incluido en el mensaje se empleará el estándar de decodificación MIME.

3.2.2 Diseño de las Interfaces de Usuario

En esta etapa del diseño, se debe definir cómo van a interactuar los usuarios con el sistema. Aquí se deben emparar las tareas humanas y orientadas a la computadora a fin de lograr el cumplimiento de los requerimientos definidos para el sistema. Esta actividad está más enfocada al estudio de las personas que de la tecnología, ya que hay que determinar, ¿Quiénes son los usuarios finales del sistema? ¿Nivel de conocimientos de la semántica y sintaxis del área de aplicación? ¿Cómo interpretan la información proporcionada por el sistema? ¿Qué es lo que esperan del sistema? Estas son, entre otras tantas, algunas de las preguntas que deben responderse como parte del diseño de las interfaces de usuario.

En ocasiones se incurre en el error de relegar este punto o no darle la importancia necesaria, sin considerar que es determinante para el éxito o fracaso del sistema. Si no es considerado en esta etapa del diseño, deberán hacerse iteraciones innecesarias, se tendrán retrasos conforme a lo planeado, resultando en la frustración del cliente.

Por tanto para el diseño de las interfaces de usuario se deben considerar los siguientes puntos:

- a) Debe ser lo más “amigable” posible,
 - Proporcionar la ayuda necesaria sobre el funcionamiento de preferencia en línea, lo cual reduce el tiempo requerido por el usuario para obtener la información.
 - Facilidad de navegación dentro del sistema. Proporcionar accesos rápidos o atajos para poder acceder a alguna opción, facilidad para desplazarse entre las opciones del menú o a través de las ventanas que conforman al sistema.
 - Los mensajes de error o de advertencia deben ser lo más claro posibles, orientar al usuario sobre que hacer en caso de presentarse alguno y nunca por ningún motivo, hacer sentir al usuario como el culpable de la ocurrencia del mismo.

- b) Tiempo respuesta, es la medida desde que el usuario realizó alguna acción de control hasta que el sistema respondió con la salida o acción esperada. Este punto tiene dos características importantes, longitud y variabilidad.
- Longitud. Es la dimensión del tiempo transcurrido desde la solicitud del usuario hasta que se obtiene la respuesta del sistema. Si el tiempo de respuesta es demasiado largo, se tendrá la desesperación del usuario.
 - Variabilidad. Corresponde con la desviación sobre el tiempo promedio de respuesta del sistema. Una baja variabilidad en el tiempo de respuesta, permite al usuario establecer un promedio en el tiempo de respuesta.

En la actualidad existen herramientas para el desarrollo de Interfaces de Usuario Gráficas (GUI, *Graphic User Interface*) que facilitan la creación de menús, ventanas, mensajes de error, interacción con dispositivos, ayudas en línea, etcétera; que normalmente involucran el empleo de principios y modelos para análisis y diseño orientado a objetos.

En cuanto a cómo lograr un buen diseño de una GUI de usuario, en la bibliografía consultada, los diferentes autores coinciden en los criterios y que se pueden agrupar de la siguiente manera:

De interacción con el usuario

- Control del usuario. El usuario debe tener la libertad para desplazarse a lo largo de la aplicación (ya sea empleando el teclado o el ratón), con lo cual se abate la frustración que el empleo de sistemas de cómputo puede ocasionar en los usuarios. Así mismo, es importante notificarle cuando el sistema le retire el control.
- Personalización. Permitir al usuario modificar la manera en que la información es desplegada o seleccionar sólo la que le interesa, por ejemplo, poder modificar el ancho de la columna en una tabla que despliega el resultado de una consulta a fin de poder ver el contenido de ésta ya que con el ancho predefinido no es posible verla.
- Solicitar confirmación de aquellas acciones destructivas o que alteren información crítica para el negocio.
- Accesos cortos. Utilizar combinación de teclas (teclas aceleradoras) que faciliten al usuario el acceder a alguna ventana o información sin necesidad de navegar a lo largo de los menús y ventanas para acceder a ella.
- Proporcionar ayudas que sean sensibles al contexto.

Información desplegada

- Sensibilidad. Siempre notificar al usuario por cada acción tomada o realizada por éste, como desplegar una barra de avance cuando el sistema esté realizando una tarea que tome algún tiempo para su ejecución.
- Consistencia. La aplicación debe ser consistente en cuanto su apariencia y sensibilidad.
- Claridad. Desplegar mensajes que tengan significado real y entendible por el usuario. ¿Cuántas veces nos hemos encontrado con mensajes como el siguiente **“FATAL ERROR, 30”** cuando ocurre una situación de error?
- Desplegar sólo la información correspondiente al contexto actual.
- Estéticamente agradables.

Introducción de datos

- Minimizar las acciones de entrada solicitadas al usuario.
- Desactivar los comandos que sean innecesarios para el contexto actual.
- Mantener la consistencia entre la información desplegada y la solicitada por el sistema.
- Proporcionar la ayuda suficiente para asistir al usuario con las acciones de entrada requeridas por el sistema.

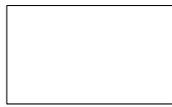
En cuanto a la interfaz del sistema con la entidad externa operador, ésta se realizará empleando el prototipo definido en la etapa de análisis, al cual se le harán las adecuaciones para incorporar los criterios citados.

3.3 Diseño Procedimental

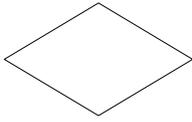
Como se mencionó, las cartas estructuradas sólo definen la jerarquía de control, pero no la secuencia de procesamiento ni las decisiones, el diseño procedimental se centra en los detalles de procesamiento de cada módulo.

El diagrama de flujo es el gráfico que detalla el procesamiento realizado dentro del módulo, indicando la secuencia, los puntos de decisión y las operaciones repetitivas.

La notación básica empleada en un diagrama de flujo es la siguiente:



Tarea, proceso o función



Decisión



Flujo



Proceso subordinado o subrutina

Con los cuales se representan la secuencia, los puntos de decisión y las operaciones repetitivas (figura 3.3.1 a la figura 3.3.5).

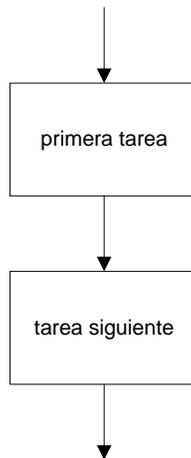


figura 3.3.1 Secuencia

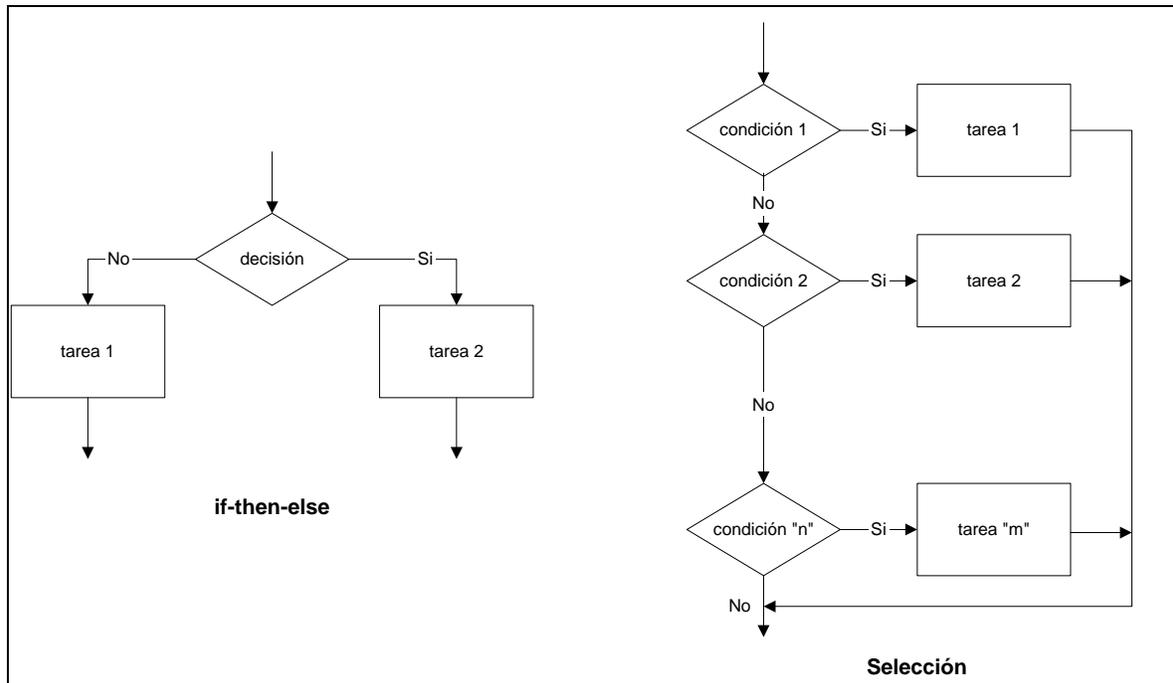


figura 3.3.2 Decisión

A continuación se presentan los diagramas de flujo para los módulos representados en las cartas estructuradas del punto 3.1.2

El diagrama de flujo obtenido para la carta estructurada del *Control de Recepción de Archivos por Correo Electrónico* se muestra en la figura 3.3.3, donde se muestra que los procesos definidos originalmente requieren un nivel mayor de detalle a fin de lograr una especificación más clara de los componentes a desarrollarse.

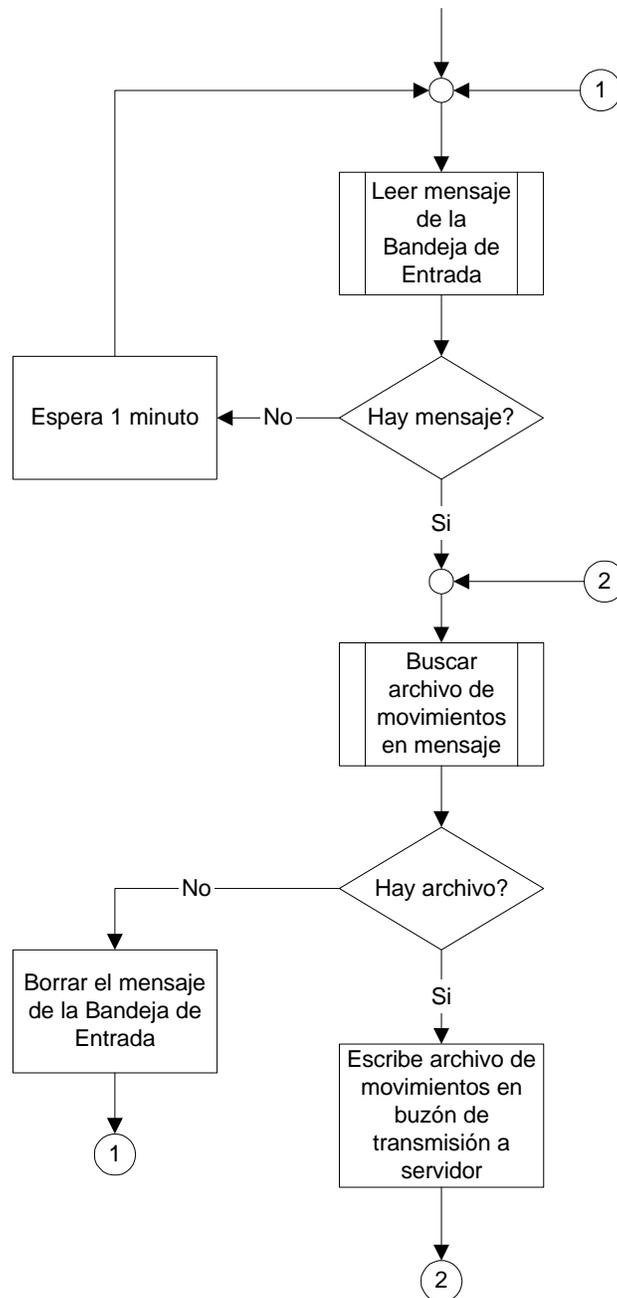


figura 3.3.3 Diagrama de Flujo Control de Recepción de Archivos por Correo Electrónico

Del diagrama anterior se desprenden los diagramas de la figura 3.3.4 y la figura 3.3.5 detallan los procesos que aparecen como subrutinas, *Leer los Mensajes de la Bandeja de Entrada* y para *Buscar archivos de Movimientos Adjuntos al Mensaje*, respectivamente.

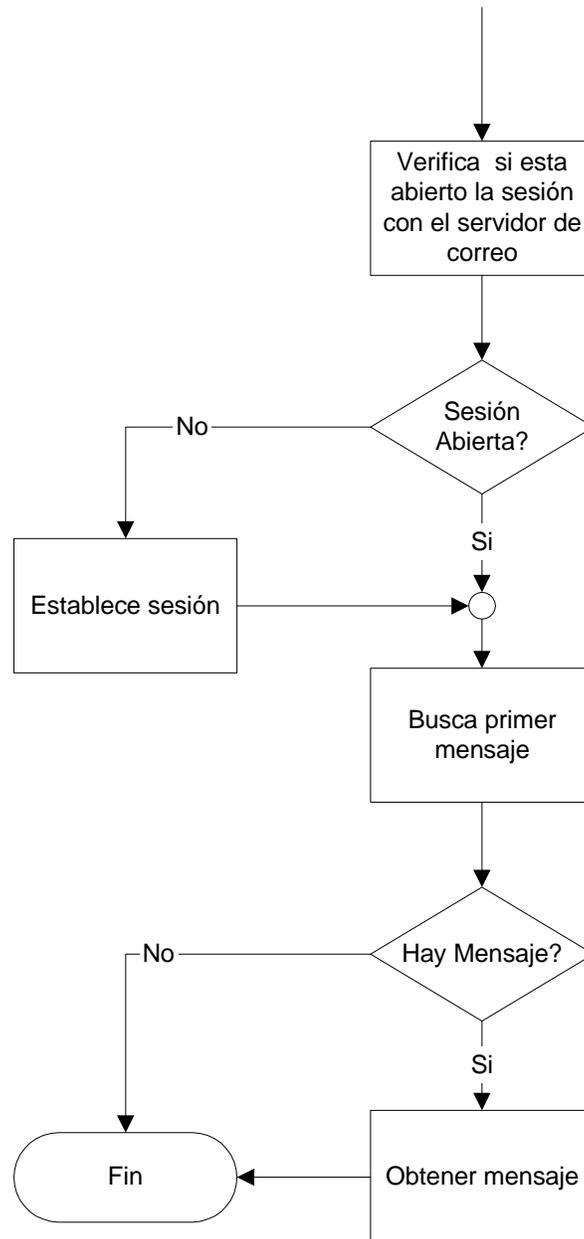


figura 3.3.4 Diagrama de Flujo Leer Mensaje de la Bandeja de Entrada

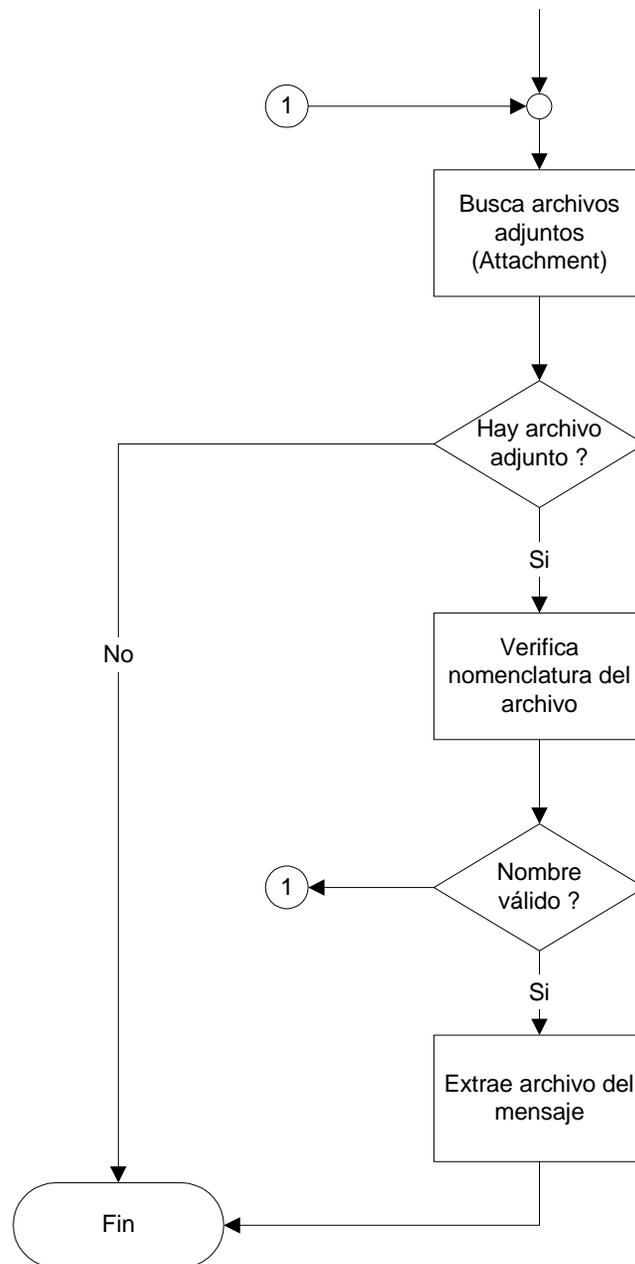


figura 3.3.5 Diagrama de Flujo Buscar Archivo de Movimientos en Mensaje

3.4 Procedimientos de Seguridad y de Control

Se diseña el procedimiento para validar o controlar la información procesada dentro del sistema, así mismo, se estructuran los procedimientos para asegurar la integridad física (acceso físico).

3.4.1 Seguridad

Encriptación

Se definen los controles criptográficos para proteger la confidencialidad, autenticidad e integridad de la información más sensible de los sistemas.

La encriptación¹ (encriptación) es utilizada como un mecanismo para “disfrazar” la información sensible o confidencial y que ésta no sea visible a todo el público, empleando para ello diferentes técnicas (criptografía) basadas en algoritmos matemáticos, para hacer inaccesible la información a personas no autorizadas. Por lo general, la encriptación se basa en una clave, sin la cual la información no puede ser descifrada.

En la actualidad existen varios algoritmos de criptografía siendo uno de los más difundidos el basado en la tecnología asimétrica originalmente definida por Diffie and Hellman, donde la encriptación - descifrición involucra diferentes llaves, una pública y otra privada; cada una con la capacidad de encriptar o descifrar datos. Un usuario entrega su llave pública a cualquier otro usuario con el cual pretenda intercambiar información, quedándose con su llave privada. Datos encriptados con la llave pública sólo pueden ser descifrados con su correspondiente llave privada y viceversa. Esta tecnología es conocida como criptografía de llave pública (Public-Key Cryptography, ver figura 3.4.1 y figura 3.4.2).

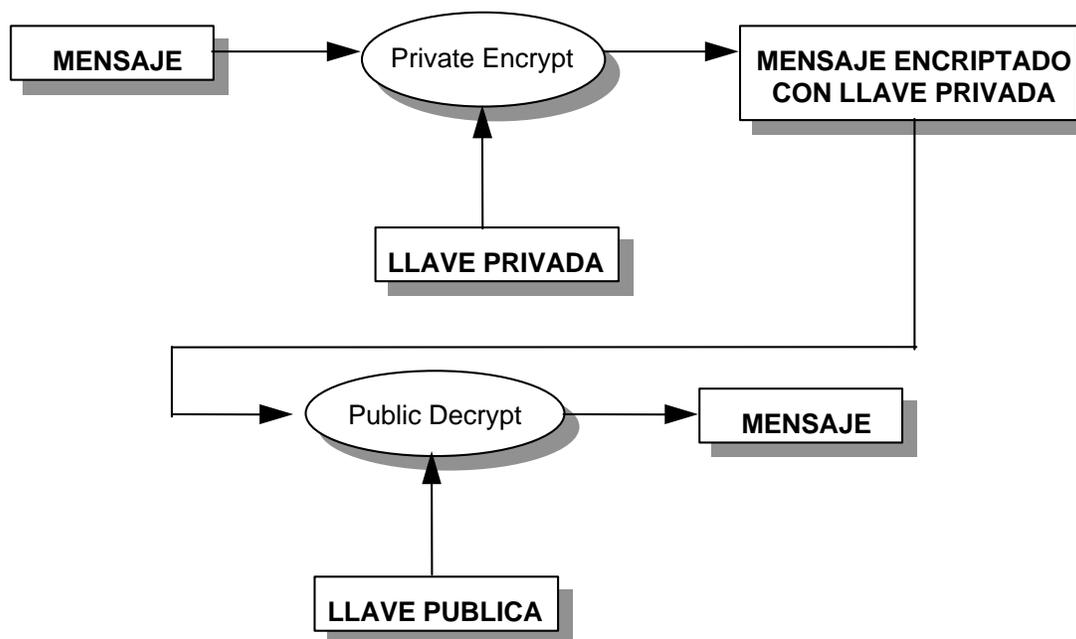


figura 3.4.1 Proceso de Encriptación con Llave Privada - Descifrición con Llave Pública

¹ De su inadecuada adaptación al español de la palabra ing. *encryption* – **codificación o encriptación**

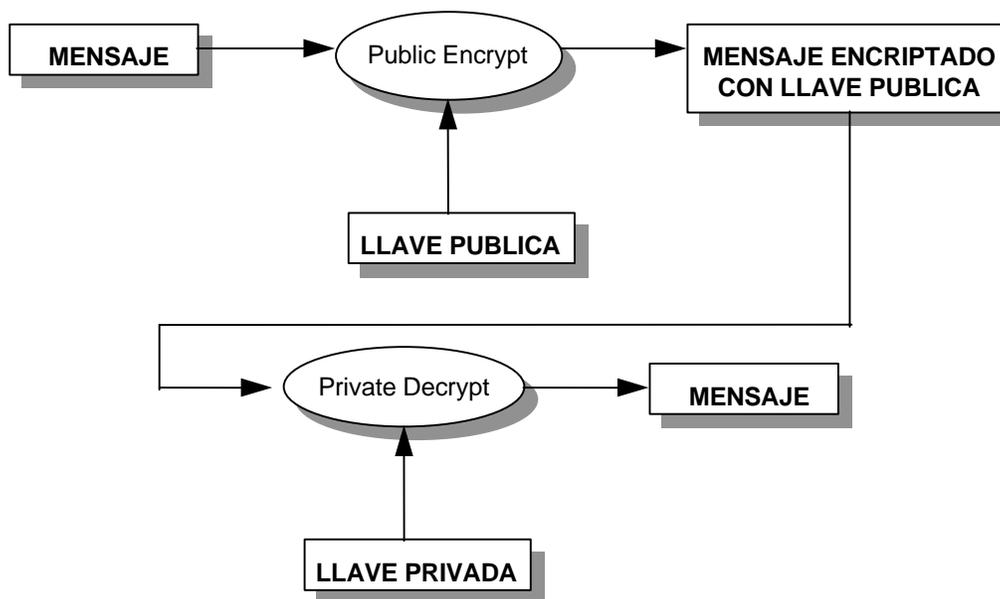


figura 3.4.2 Proceso de Encriptación con Llave Pública - Desencriptación con Llave Privada

Como se puede ver, esta técnica nos permite intercambiar información con la certidumbre que ésta no esté disponible a personas no autorizadas, pero no garantiza la procedencia o si la información original a sufrido modificaciones desde su origen hasta el destino, lo que nos introduce a los siguientes conceptos:

Digestión de mensaje (resumen). El algoritmo de digestión de mensajes es un método para reducir un mensaje de cualquier longitud a una cadena de longitud fija conocida como digestión o resumen, de tal manera que es imposible tener dos mensajes diferentes con la misma digestión. Si un mensaje m con una digestión d es alterado, el cual sería un mensaje diferente de m , al calcular su digestión por ende sería diferente de d . Por lo tanto, la digestión de mensaje proporciona un método que permite verificar la integridad de la información (figura 3.4.3).

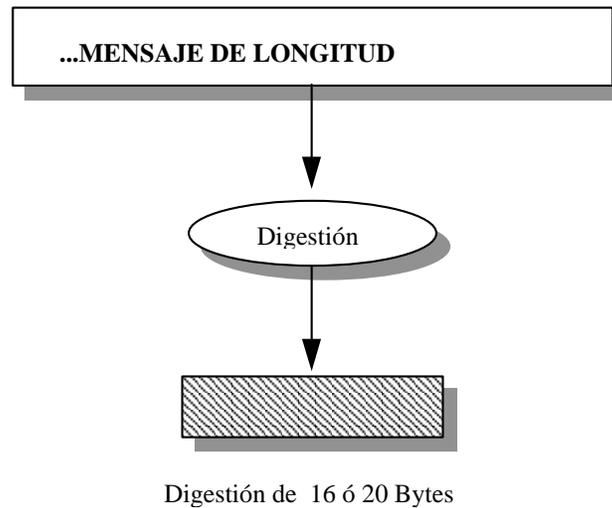


figura 3.4.3 Digestión de Mensaje

Firma Digital. La firma digital es un concepto bajo el cual un firmante, llamémosle “Alicia”, firma un mensaje m de tal manera que cualquiera pueda verificar que dicho mensaje fue firmado por “Alicia” y nadie más, y consecuentemente el mensaje no ha sido modificado desde que ella lo firmó.

La típica implementación de la firma digital involucra un algoritmo de digestión de mensaje (resumen), y un algoritmo de encriptación con llave privada.

- “Alicia” reduce el mensaje m a una digestión d con algún algoritmo de digestión de mensaje, después ella encripta la digestión d con su llave privada, obteniendo una digestión encriptada σ . Ella envía a “Beto” el mensaje m y la digestión encriptada σ . Juntos conforman un mensaje firmado digitalmente.
- “Beto” descripta la digestión encriptada σ empleando la llave pública de “Alicia” obteniendo la digestión d , posteriormente reduce el mensaje m a una digestión d' y la compara con d . Si son iguales se acepta el mensaje.

La figura 3.4.4 muestra el proceso para la generación de una firma digital.

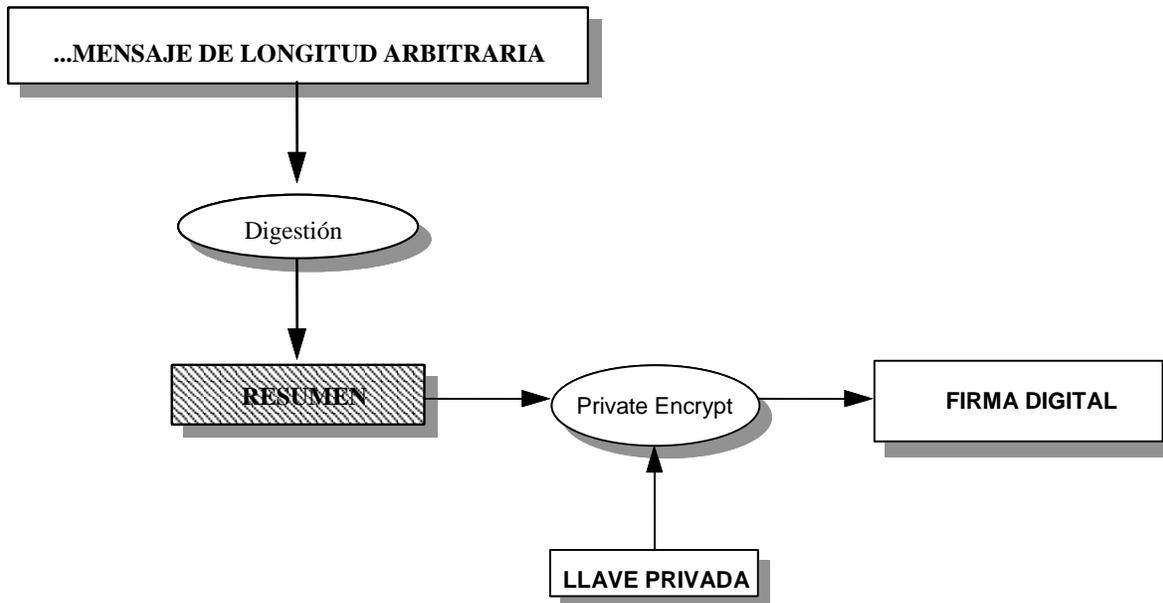


figura 3.4.4 Proceso de Generación de la Firma digital

Sobre Digital. Es una aplicación en la que alguien “sella” un mensaje m de tal manera que sólo el destinatario puede abrirlo.

La típica implementación del algoritmo de sobre digital involucra un algoritmo simétrico de llave secreta para encriptar el contenido del mensaje y otro para de llave pública para encriptar la llave secreta. Lo anterior se realiza de esta manera para hacer más eficiente la encriptación, ya que los algoritmos asimétricos son más lentos que los simétricos:

- “Alicia” encripta el mensaje m con una llave secreta k obtenida aleatoriamente, obteniéndose un mensaje encriptado c , después ella encripta la llave secreta k , con la llave pública de “Beto”, obteniendo la llave secreta encriptada σ
- “Beto” desencripta la llave secreta σ empleando su llave privada, obteniendo k , después el desencripta el mensaje encriptado c con la llave secreta k , obteniendo el mensaje m .

Los procesos para la generación y apertura de un sobre digital se muestran en la figura 3.4.5.

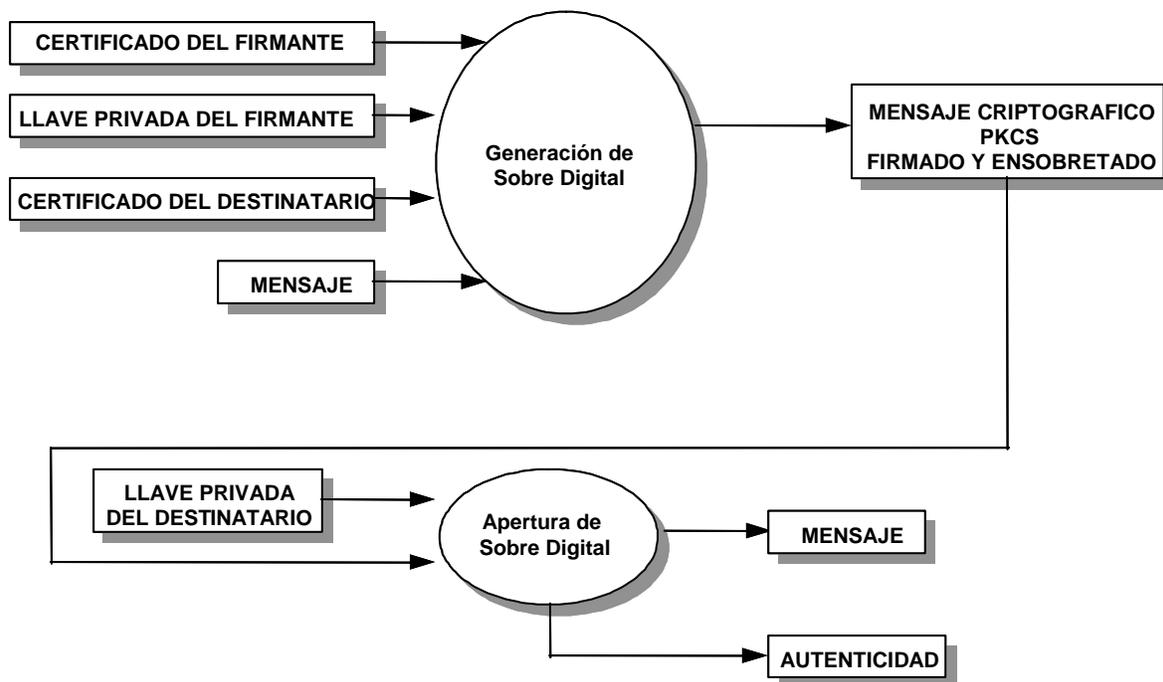


figura 3.4.5 Procesos de Generación y Apertura de Sobre Digital

Como se mencionó, existen varias tecnologías de encriptación, pero para poder hacer un intercambio seguro entre sistemas abiertos es necesario emplear los estándares existentes, siendo PKCS (*Private-Key Cryptography Standard*) de los Laboratorios RSA el más utilizado.

Para la encriptación de los archivos de movimientos y de respuesta intercambiados entre los recintos y la SHCP, se seleccionó el empleo de los sobres digitales definidos en el estándar PKCS #7. Lo cual implica que a cada Recinto Fiscalizado le será proporcionado su par de llaves, así como la llave pública de la SHCP, para poder intercambiar la información. Considerando que la SHCP es una autoridad certificadora.

Dentro del servidor UNIX serán abiertos los sobres digitales conteniendo los archivos de movimientos y serán generados los sobres digitales conteniendo los archivos de respuesta. El flujo completo está representado en la figura 3.4.6.

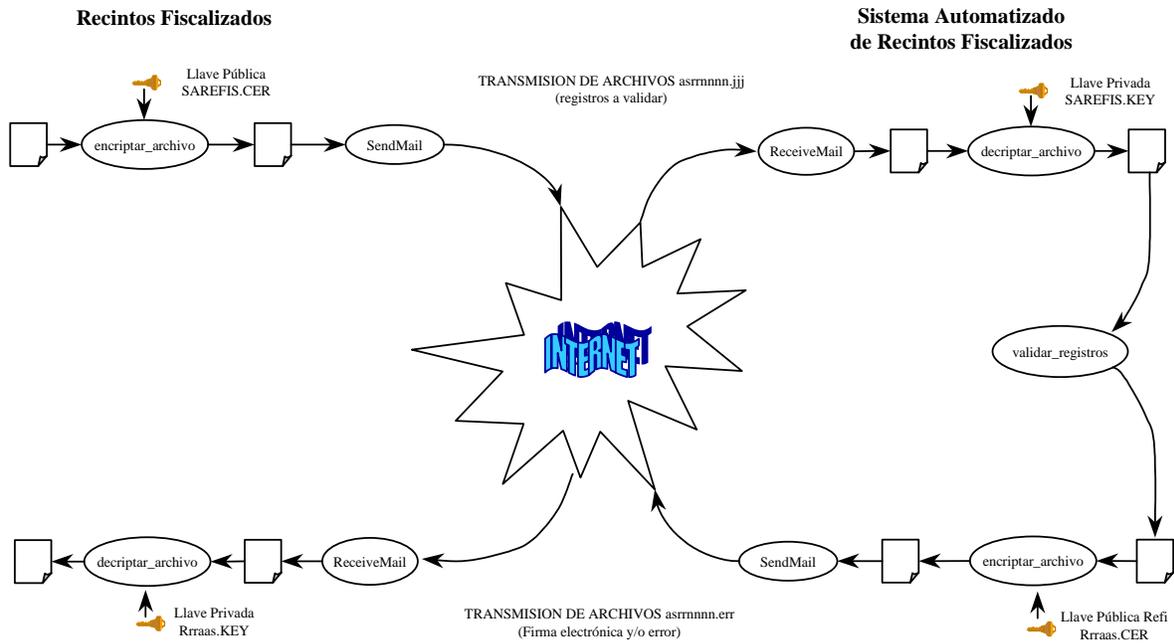


figura 3.4.6 Procesos de Transferencia de Archivos, Apertura y Generación de Sello Digital

En las figuras 3.4.7 y 3.4.8 se muestra el refinamiento realizado a la Carta Estructurada y al DTE del Control de la Validación para incorporar las funciones para Obtener el Certificado, Abrir y Generar Sobre Digital.

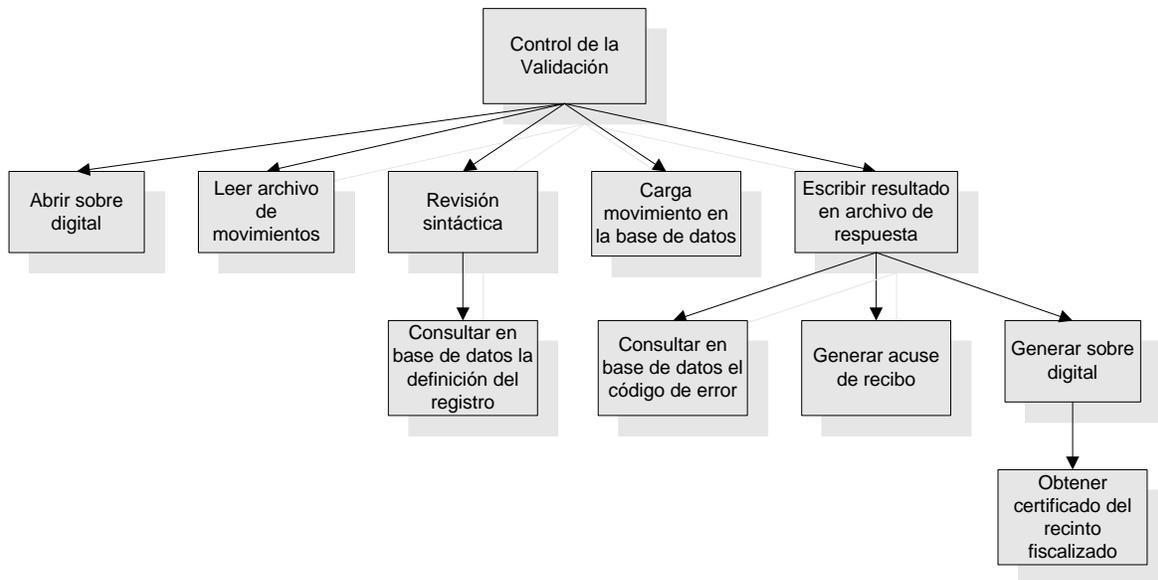


figura 3.4.7 Carta Estructurada Refinada del Control de la Validación

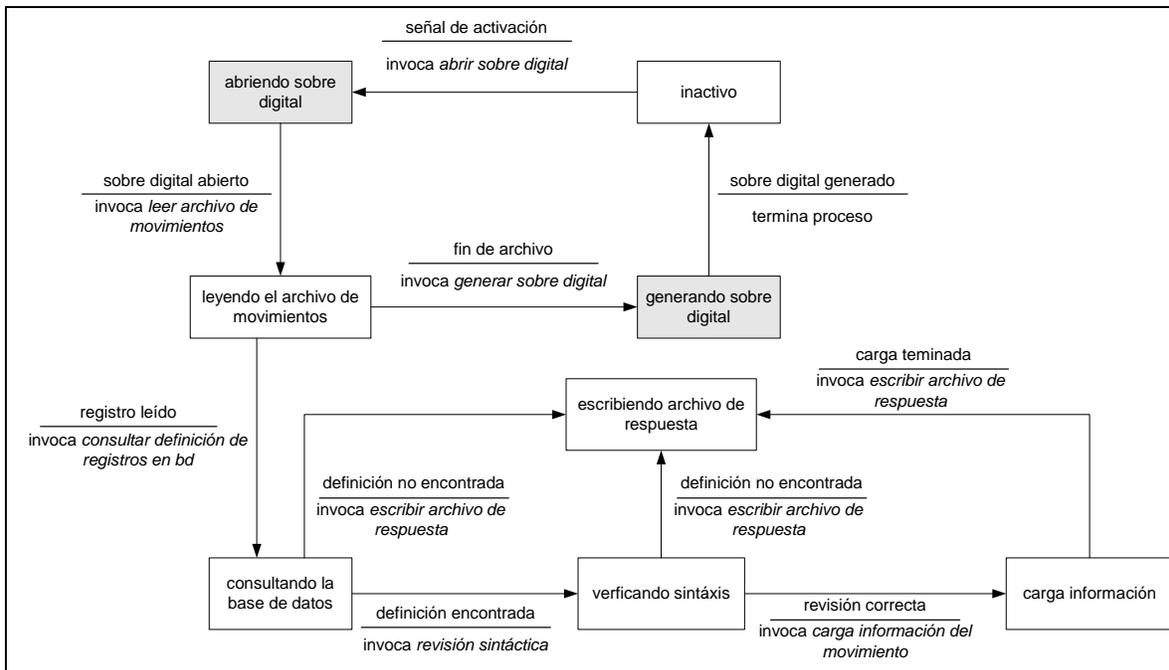


figura 3.4.8 DTE Refinado del Control de la Validación

3.4.2 Control

Se definen los mecanismos a emplearse para restringir el acceso de los usuarios a la información y a las operaciones que puede realizar sobre ella, así como a las opciones del sistema y recursos que sólo le han sido autorizados.

Controles para el servidor UNIX:

- La información empleada por el sistema se almacena en la base de datos dentro del servidor UNIX, utilizándose INFORMIX como sistema administrador de base de datos (DBMS), el cual emplea la validación de accesos definida por el sistema operativo (SO).
- Se creará un usuario con los privilegios necesarios para conectarse a la base de datos y realizar operaciones de SELECT, INSERT, UPDATE, DELETE y EXECUTE. No debe tener acceso al shell del SO, sólo se define para poder ejecutar los programas que se ejecutan en "batch" y los servicios de la aplicación.
- Los programas tendrán sólo permisos de ejecución para el propietario, a fin de que ningún otro usuario pueda ejecutarlos indebidamente.
- Se define un puerto para la ejecución de los servicios del sistema (de comandos SQL, habilitados sólo para las direcciones IP de los equipos cliente previamente autorizados).

Controles para el Servidor Windows NT y Clientes:

- Se creará un usuario con los privilegios adecuados para acceder al buzón de correo electrónico y de ejecución de los servicios.
- Los servicios para el envío y recepción de correos y archivos movimientos serán configurados para que sean ejecutados por el usuario creado.
- La validación de acceso para los usuarios a la aplicación cliente (front-end), se hará a través de un identificador de usuario y contraseña, los cuales serán almacenados encriptados en la base de datos del servidor UNIX.
- Se definen perfiles para restringir los accesos a las opciones y operaciones autorizadas para éstos como se muestra en la figura 3.4.9.

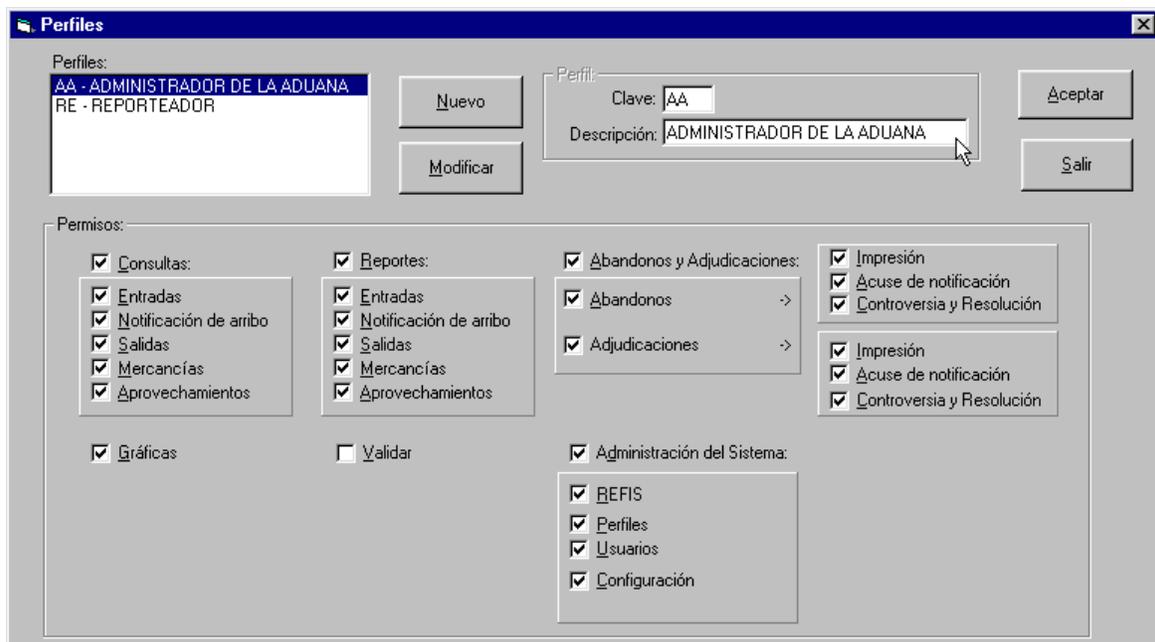


figura 3.4.9 Opción de Perfiles del Menú de Administración

- La administración de los usuarios y perfiles se realiza desde el front-end por un usuario con privilegios de administrador. El administrador tiene la libertad para generar los perfiles conforme más convenga a la operación de la aduana.
- Todos los usuarios del sistema tienen la posibilidad de ingresar su contraseña y modificarla cuando lo deseen si esta opción se encuentra habilitada para el perfil al que pertenece (figura 3.4.10).

figura 3.4.10 Opción de Usuarios del Menú de Administración

3.5 Diseño del Plan de Pruebas

En el diseño del plan de pruebas desarrollan y documentan los indicadores que permitan evaluar los aspectos de funcionamiento del sistema (casos de prueba) y como aplicarlos.

Las pruebas es un elemento vital para el aseguramiento de la calidad del sistema desarrollado ya que representa la última revisión de las especificaciones, diseño y construcción antes de su liberación e implantación. Siendo un proceso que tiene como finalidad el descubrir todos los errores existentes al ejecutar un programa, es decir, una buena prueba es aquella que descubre los errores que hasta el momento de aplicarla no habían sido encontrados.

Como parte del diseño de los casos de prueba hay que considerar los siguientes principios:

- Todas las pruebas deben conducir a los requerimientos del cliente. Considerando que la intención es descubrir errores, el no cubrir los requerimientos o especificaciones del usuario debe verse como un defecto.
- Las pruebas a realizarse deben ir de lo particular a lo general. Para iniciar, deben enfocarse en módulos individuales (pruebas unitarias) y conforme se avanza en las pruebas, se pretenderá encontrar errores en el sistema completo (pruebas integrales).
- Resulta casi imposible probar todo. Las diferentes rutas o caminos de ejecución que cada módulo pueda seguir, por sencillo que éste sea, puede

hacer muy extensa la realización de las pruebas del sistema. Es por ello que se deben diseñar los casos de prueba de manera que cubran adecuadamente la lógica y se ejecuten todas las condiciones reflejadas en el diseño procedimental.

- Para que las pruebas sean efectivas, deben ser aplicadas por un tercero. El desarrollador no es la persona más indicada para aplicar las pruebas, pues tomando en cuenta que no es natural ejecutar el sistema con la intención de encontrar errores (“hacerlo pedazos”), lo cual resulta ser una actividad destructiva.
- La mejor prueba no es la que nos dice lo bien que funciona el sistema, sino aquella que descubre todos los errores presentes en él.

3.5.1 Casos de Prueba

Para el diseño de los casos de pruebas de un sistema, existen dos enfoques:

Prueba de caja negra, orientada a verificar la funcionalidad y comportamiento observable para la cual el sistema fue diseñado, a fin de descubrir errores de funcionalidad sin considerar el comportamiento interno de cada módulo o como fue implementado. Se centra en el dominio de la información del sistema. Se deben generar los casos de prueba necesarios para verificar lo siguiente:

- Por cada valor válido empleado como entrada, se obtuvo un valor apropiado como resultado.
- Por cada valor inválido empleado como entrada, se obtuvo un sólo valor apropiado como resultado.
- Por cada estado de entrada válido, se obtuvo un estado apropiado de salida.
- Por cada estado de entrada inválido, se obtuvo un estado apropiado de salida.

Prueba de caja blanca (o de cristal), encaminada a probar todas las operaciones internas con base en las especificaciones de diseño. Normalmente se aplica a pequeños componentes o unidades del sistema. Teóricamente se deben probar todos los flujos posibles a través del código, pero esto sólo es posible en módulos o unidades muy simples. Los casos de prueba generados deben ejecutar cada uno de los flujos de decisión con lo cual se ejecuta al menos una vez todas las sentencias

3.5.2 Tipos de Pruebas

Considerando a las pruebas como actividad esencial para el aseguramiento de la calidad, es necesario detectar cualquier error existente dentro del sistema por lo

cual es necesario aplicar varias pruebas encaminadas a detectar los diversos tipos de errores en el funcionamiento del sistema.

Básicamente los tipos de pruebas aplicables son:

- **De sentencias.** Se aplican con la intención de detectar errores en la ejecución de las rutas especificadas en los diagramas de diseño.
- **De volumen.** Es importante probar como se comporta el sistema cuando es sometido a grandes volúmenes de información.
- **De esfuerzo (stress).** En la realidad, los sistemas normalmente son sometidos a cargas de trabajo que no son uniformes, existen momentos del día o fechas en que la cantidad de procesos ejecutados, usuarios que acceden al sistema, etcétera, se ven incrementadas, es por ello que es necesario verificar el comportamiento del sistema ante estas circunstancias, como responde ante la concurrencia y como se degrada el desempeño, entre otros aspectos.
- **De facilidad de uso.** Esta es una premisa que debe comprobarse durante las pruebas, ya que de esto dependerá el nivel de aceptación por parte del cliente y usuario final.
- **De seguridad.** Se centra en detectar la vulnerabilidad del sistema ante los accesos a la información y procesos de la aplicación, conforme a los requerimientos de seguridad establecidos.
- **De comportamiento.** Se debe evaluar como se desempeñó el sistema. Medir los tiempos requeridos por el sistema para responder a las acciones solicitadas
- **De almacenamiento.** Es necesario comprobar que la información es almacenada y/o actualizada en las estructuras definidas para ello, que se respeta la integridad de los datos, las referencias, etcétera.
- **De configuración.** Se aplican con la intención de detectar como se comporta el sistema ante diferentes configuraciones.
- **De facilidad de instalación.** Se comprueba la complejidad para instalar el software, la cual debe de ir acorde con el nivel técnico del usuario el cual la va a realizar.
- **De confiabilidad.** El sistema siempre debe responder adecuadamente ante cualquier acción realizada por el usuario por descabellada que ésta sea, ante cualquier captura de datos fuera de los dominios definidos, es decir, a prueba de fallas y en caso de presentarse un error grave, cómo actúa ante éste y si puede recuperarse del mismo.

3.5.3 Niveles de Pruebas

Las pruebas son aplicadas sobre diferentes tipos de objetivos en diferentes etapas o niveles de esfuerzo. Estos niveles se distinguen básicamente por aquellos roles con las mejores habilidades para diseñar y conducir las pruebas.

- **De desarrollador.** Son aquellas diseñadas y llevadas a cabo por el equipo de desarrolladores, generalmente se les asocia con las pruebas unitarias aunque en ocasiones, existen desarrolladores que participan realizando pruebas de integración.
- **Independientes.** Aquellas diseñadas y llevadas a cabo por un equipo independiente al de desarrolladores, las cuales incluyen una verificación y validación independiente. El propósito de las pruebas independientes es proveer de una perspectiva diferente y diferentes casos de prueba que se realicen en un ambiente más rico del que es posible para el desarrollador.
- **Unitarias.** Se centran en probar las unidades más pequeñas del software, normalmente aplicadas para verificar que los flujos de control y de datos son cubiertos y que funcionan como se espera
- **De integración.** Se realizan para asegurar que al integrar los componentes operan adecuadamente.
- **De sistema.** Son realizadas una vez que el sistema funciona como un todo, enfocándose en el funcionamiento de los elementos de principio a fin.
- **De aceptación.** Son las pruebas finales realizadas antes de distribuir el software. La finalidad es verificar que el software está listo, y que puede ser usado por los usuarios para realizar las tareas para lo cual fue desarrollado.

La siguiente matriz es un ejemplo de algunos de los casos de prueba definidos para el sistema y cómo aplicarlos. Para el ejemplo se definieron casos de prueba de caja negra desde la perspectiva del equipo de desarrollo.

Casos de Pruebas

Consideraciones	Caso	Funcionalidad verificada	Resultados esperados
Generar varios archivos de movimientos conteniendo los diferentes tipos de registro, "ensobretarlos" digitalmente empleando una llave privada proporcionada por el SAT y enviarlos por correo electrónico a la dirección de correo de la aplicación.	1. Emplear valores válidos para la generación de los registros, en cuanto a formato, rangos, valores, secuencia, etcétera. Los archivos deben ir adjuntos al correo enviado. Enviar archivo por archivo después de recibir respuesta del anterior.	Comprobar configuración del buzón de correo de la aplicación	Se deben recibir los correos enviados.
		Verificar la funcionalidad del servicio de Recepción de Correo Electrónico	1. Se debe extraer los archivos adjuntos y depositarlos en la carpeta de envío al servidor. 2. El correo debe ser eliminado del buzón.
		Verificar la funcionalidad del servicio de Transmisión de Archivos al Servidor	Los archivos de movimientos deben aparecer en la carpeta de entrada para validación en el servidor UNIX.
		Verificar la funcionalidad del Validador	1. Los archivos deben ser descryptados y leídos por el validador. 2. Se debe generar el archivo de respuesta conteniendo acuse electrónico para cada uno de los movimientos reportados. Este archivo debe estar "ensobretado digitalmente"
		Verificar la funcionalidad del servicio de Recepción de Archivos.	Los archivos de respuesta deben aparecer en la carpeta de recepción en el servidor NT.
		Verificar la funcionalidad del servicio de Envío de Correo Electrónico	Los archivos de respuesta deben ser entregados a la dirección de correo registrada para el recinto que envió la información.

Consideraciones	Caso	Funcionalidad verificada	Resultados esperados
		Verificar el almacenamiento de los datos.	Los movimientos se deben almacenar en las tablas y campos correspondientes. Con los valores declarados.
	2. Emplear valores válidos para la generación de los registros, en cuanto a formatos, rangos, valores, secuencias, etcétera. Los archivos deben ir adjuntos al correo enviado. Enviar varios archivos a la vez.	Las mismas para el caso 1.	Los mismos para el caso 1.
	3. Emplear valores válidos para la generación de los registros, en cuanto a formatos, rangos, valores, secuencias, etcétera. Los archivos deben ir adjuntos al correo enviado. Enviar archivo por archivo después de recibir respuesta del anterior. Los archivos deben contener gran número de movimientos.	Las mismas para el caso 1.	Los mismos para el caso 1. El tiempo de respuesta debe crecer proporcionalmente al número de movimientos contenidos en el archivo.
	4. Emplear valores no válidos para la generación de los registros, en cuanto a formatos, rangos, valores, secuencias, etcétera. Los archivos deben ir adjuntos al correo enviado. Enviar archivo por archivo después de recibir respuesta del anterior.	Verificar la funcionalidad del Validador	<p>1. Los archivos deben ser descryptados y leídos por el validador.</p> <p>2. Se debe generar el archivo de respuesta conteniendo los errores para cada uno de los registros dependiendo de los tipos de valores no validos empleados. Este archivo debe estar "ensobretado digitalmente"</p>

Consideraciones	Caso	Funcionalidad verificada	Resultados esperados
	5. Emplear valores válidos para la generación de los registros, en cuanto a formatos, rangos, valores, secuencias, etcétera. Los archivos deben ir adjuntos al correo enviado con nomenclatura incorrecta. Enviar archivo por archivo después de recibir respuesta del anterior	Verificar la funcionalidad del servicio de Recepción de Correo Electrónico	1. No se debe extraer los archivos. 2. El correo debe ser eliminado del buzón.
	6. Remover o renombrar el archivo de Certificado correspondiente en la carpeta de CERTIFICADOS del servidor UNIX.	Verificar la funcionalidad del Validador	Los archivos deben ser depositados en la carpeta de Rechazados al no poderse verificar su procedencia.

3.6 Diseño del plan de transición

En él se definen las estrategias que permitan minimizar los problemas que puedan surgir durante la implantación del sistema y su puesta en operación. La instalación de un nuevo sistema puede ofrecer un gran cambio y generar estrés en los empleados del cliente, es por ello que debe considerar minimizar el impacto que el sistema pueda presentar sobre los procesos del negocio y su personal.

3.6.1 Estrategia de transición

Se define la estrategia que permita prever como será el arranque de sistema, tomando en cuenta los requerimientos definidos.

Así mismo, si el sistema a desarrollarse sustituirá a otro ya existente hay considerar las actividades relacionadas con la compatibilidad, conversión y migración, específicamente:

- Los datos del sistema existente deben ser cargados en el nuevo y probablemente convertidos a otro formato.
- Las interfaces de usuario existentes deben ser soportadas por el nuevo sistema.
- Las interfaces de aplicación (APIs) deben conservarse.
- El nuevo sistema debe tener la capacidad de operar en paralelo con el existente durante el período de migración.
- Debe ser capaz de permitir regresar la operación al sistema existente si fuese necesario.

Para el sistema a desarrollarse esta situación no aplica ya que es un sistema nuevo.

3.6.2 Diseñar el plan de implantación del sistema

En el plan de implantación debe hacerse énfasis en la incorporación de los módulos o servicios completos que comprenden las instalaciones, hardware y software necesario.

Algunos sistemas deben ser liberados de manera incremental, en partes o disponibles en alguna fecha. Si el sistema no puede ser instalado todo en una sola vez, se debe establecer el orden en que los componentes deben ser instalados y en que nodos. Algunos de los patrones más comunes para los planes de implantación son:

- Geográfico, por localidad.
- Funcional, por aplicación.
- Organizacional, por departamento o área de negocio.

Cuando una aplicación es liberada durante un período de tiempo deben tomarse en cuenta los siguientes aspectos:

- El software debe poderse ejecutar en una configuración parcial.
- Diferentes versiones de la aplicación deben poder coexistir.
- Debe ser posible regresar a una versión previa del sistema cuando un problema o falla así lo amerite.

En el caso de la SHCP y por su tipo de operación existe una división organizacional por cada área de negocio o proceso sustantivo, comercio exterior, recaudación, auditoría fiscal, etcétera, por lo que los nodos de cada uno de ellos se encuentran separados y perfectamente identificados.

Hablando del área de negocio de comercio exterior, las aduanas del país están clasificadas por su ubicación y tipo de operación en fronteriza, marítimas, aeroportuarias e interiores. Donde cada una de ellas cuentan con equipos de diferentes plataformas, UNIX e Intel.

En el caso de aduanas, debido a la importancia que la operación de los sistemas tiene para el comercio exterior del país, cualquier falla tiene consecuencias de alto impacto para todos los actores que en ella participan (aduanas, agentes aduanales, importadores, SHCP, etcétera), es necesario realizar la liberación e instalación de los sistemas en dos etapas, para la primera se selecciona la localidad o aduana que fungirá como piloto, a fin de verificar la correcta funcionalidad de la aplicación así como su desempeño en un ambiente de producción, una vez que es aprobado y aceptado por los usuarios, se realiza la liberación a todas las demás aduanas. En caso de ser necesario, se puede extender el piloto a otra(s) con diferente tipo de operación hasta cubrir cada uno de ellos antes de su implantación a nivel nacional. Para seleccionar la aduana a emplearse como piloto, se considera que el tipo de operación de ésta concuerde con la funcionalidad de la aplicación liberada, además se debe considerar el volumen de operación y en las fechas que éstas son mayores.

Por otro lado, es necesario mencionar si el sistema o aplicación a liberarse tiene requerimientos previos de instalación, si detiene la operación de algún módulo en específico o la operación de toda la aduana para su instalación, así como su dependencia con otras liberaciones y el orden de ejecución de cada una de ellas.

A continuación en la figura 3.6.1 se muestra la carátula de liberación de software la cual reúne los requisitos necesarios para plan de implantación:

El nombre del paquete a instalarse depende del nodo y el área de negocio		Nombre del paquete	Extensión	Fecha	Prioridad	No. de solicitud del Usuario y/o reporte de Mesa de Ayuda	Folio
Finalidad del software liberado		Objetivo:					
Se especifica si es necesario detener total o parcialmente la operación de la aduana para su instalación		Sistema/subsistema	Equipos/localidades				
Comentarios complementarios para el operador que realizará la instalación del paquete		Modificación a: <input type="checkbox"/> Operación <input type="checkbox"/> Base Datos <input type="checkbox"/> Estructura <input type="checkbox"/> Catálogos <input type="checkbox"/> Programa <input type="checkbox"/> Archivos <input type="checkbox"/> Estructura Directorios					
REQUERIMIENTOS PREVIOS A LA INSTALACIÓN DEL PAQUETE							
Suspender operación		NO					
Demonios / automático		No aplica					
Base de Datos		Estado: NO APLICA			Respaldar: NO APLICA		
Exclusivo Aduanas relación con otras actualizaciones		No Aplica					
Observaciones							

Determina la oportunidad que el paquete distribuido debe ser probado e instalado en producción

Documento que justifica los cambios realizados y la liberación

Proyecto dentro del control de versiones al que pertenece la programación del software liberado

Nodos y plataforma al cual debe instalarse la liberación

Indicar la dependencia con otras liberaciones y orden de ejecución

figura 3.6.1 Carátula de liberación de software

Cabe mencionar que para la implantación de los sistemas, es necesario prever como los módulos desarrollados son controlados e integrados para conformar el sistema, los nodos en que serán distribuidos, rutas, permisos y propietarios de éstos.

A continuación se muestran las matrices diseñadas a fin de poder realizar el control y liberación de los módulos desarrollados (figura 3.6.2 y figura 3.6.3):

CONTROL DE CAMBIOS			
PROYECTO/SUBPROYECTO			
INSTALABLES AFECTADOS			
CUENTA DE TRABAJO DEL USUARIO	NOMBRE	SUPERVISOR	
DIRECTORIO PARA EDICIÓN/INGRESO			
FECHA DE EXTRACCIÓN PARA EDICIÓN	FECHA DE INGRESO DE CAMBIOS		
JUSTIFICACIÓN(ES) DE LOS CAMBIOS			
ARCHIVO A MODIFICAR	VERSIÓN		DESCRIPCIÓN DE LOS CAMBIOS
	ANTERIOR	NUEVA	

figura 3.6.2 Hoja de control de cambios

Capítulo 4. Construcción

Capítulo 4. Construcción

Durante la fase de construcción, se codifican las especificaciones hechas en la etapa de diseño de acuerdo con el o los lenguajes de programación seleccionados para la implementación de la solución. Así mismo, el equipo de desarrollo debe realizar las pruebas necesarias a fin de garantizar la calidad del sistema, apoyándose en los casos de prueba definidos durante la fase de diseño.

Se muestran a continuación ejemplos del código escrito para la construcción de algunos de los módulos del sistema.

Como parte del cliente para la ejecución de comandos SQL, fue necesario construir las clases y métodos para conectarse a la base de datos y obtener la información de la misma. El siguiente código en Visual Basic muestra algunas de las clases desarrolladas:

Campo.Cls

```
'Clase Campo

'Método público para obtener el nombre del campo
Property Get Nombre() As String
    Nombre = C_Nombre
End Property

'Método público para asignar el nombre del campo
Property Let Nombre(strNombre As String)
    C_Nombre = strNombre
End Property

'Método público para asignar el tamaño o longitud del campo
Property Let Tamano(iTamano As Integer)
    C_Tamano = iTamano
End Property

'Método público para obtener el tamaño o longitud del campo
Property Get Tamano() As Integer
    Tamano = C_Tamano
End Property

'Método público para asignar el tipo de dato del campo
Property Let Tipo(strTipo As String)
    C_Tipo = strTipo
End Property

'Método público para obtener el tipo de dato del campo
Property Get Tipo() As String
    Tipo = C_Tipo
End Property

'Método público para asignar el valor del campo
Property Let Valor(strValor As String)
    C_Valor = strValor
End Property

'Método público para obtener el nombre del campo
Property Get Valor() As String
    Valor = C_Valor
End Property
```

Fin Campos.cls

Ccampos.Cls

```

'Clase Ccampos que corresponde a una colección de campos

'Definición del objeto Campos como una colección
Private Campos As New Collection

'Método público para agregar objetos del tipo Campo a la colección
Public Function Agregar(ByVal strNombre As String, _
    ByVal strValor As String, _
    ByVal strTipo As String, _
    ByVal iTamano As Integer) As Campo

    'Se declara un objeto del tipo campo
    Dim vCampo As New Campo

    'Se asignan los valores para los atributos del objeto vCampo
    With vCampo
        .Nombre = strNombre
        .Valor = strValor
        .Tipo = strTipo
        .Tamano = iTamano
    End With

    'Se atrapa el error
    On Error GoTo salida

    'Se añade el objeto a la colección, utilizando su nombre como índice
    Campos.Add vCampo, vCampo.Nombre

    'El método devuelve un objeto del tipo Campo
    Set Agregar = vCampo
    Exit Function

salida:
    Err.Raise Err.Description
    On Error GoTo 0
End Function

' Método público para eliminar un elemento de la colección
Public Sub Eliminar(Item As Integer)

    On Error GoTo salida

    'Se remueve el elemento de la colección
    Campos.Remove Item
    Exit Sub

salida:
    Err.Raise Err.Description
    On Error GoTo 0

End Sub

'Método público para conocer cuantos elementos tiene la colección
Public Function Count() As Long

    Count = Campos.Count

End Function

'Método público para obtener un elemento del tipo Campo de la colección
Public Function Item(ByVal i As Variant) As Campo

    Set Item = Campos(i)

End Function

```

Fin CCampos.cls

Sqlclin.bas

```

' --- Definiciones para el manejo de comandos SQL remotos ---

' Constantes para el desplazamiento del CURSOR
Public Const AVANZA_PRIMERO As Long = 0
Public Const AVANZA_ULTIMO As Long = 1
Public Const AVANZA_SIGUIENTE As Long = 2
Public Const AVANZA_ANTERIOR As Long = 3

' Constantes para operaciones de actualización y transacciones
Public Const BD_EXECUTE As Long = 0
Public Const BD_CERRAR As Long = 4
Public Const BD_BEGINTRANS As Long = 5
Public Const BD_COMMIT As Long = 6
Public Const BD_ROLLBACK As Long = 7
Public Const BD_ABRIR As Long = 8

' Constantes para la dirección de búsqueda
Public Const BUSCAR_PRIMERO As Long = 9
Public Const BUSCAR_SIGUIENTE As Long = 10
Public Const BUSCAR_ULTIMO As Long = 11

Public Const C_REQUERY As Integer = 12

' Declaración de funciones incluidas en la biblioteca sqlclin.dll
' para la ejecución de comandos SQL remotos empleando SOCKETS
' emplea el servicio "qrysrv 50245/tcp"

Declare Function conecta Lib "sqlclin" (ByVal strHostname As String, ByVal strServ As
String, ByRef Socket As Long) As Long

' Ejecuta comandos de consulta
Declare Function ejecuta_sql Lib "sqlclin" (ByVal Socket As Long, _
ByVal operacion As Long, ByVal qry As String, ByVal titulos As String, _
ByVal tipos As String, ByVal Tamanio As String, _
ByVal datos As String, ByRef numregs As Long) As Long

' Baja a un archivo el resultado de la consulta
Declare Function baja_a_archivo Lib "sqlclin" (ByVal lSocket As Long, _
ByVal strQry As String, ByVal strArchivo As String, _
ByVal datos As String) As Long

' Avanza a lo largo de un cursor
Declare Function avanza Lib "sqlclin" (ByVal Socket As Long, _
ByVal direccion As Long, ByVal datos As String) As Long

' Efectua comandos de actualización y transacciones
Declare Function opera_bd Lib "sqlclin" (ByVal Socket As Long, _
ByVal codigo As Long, ByVal qry As String, ByVal strErrMsg As String) _
As Long

' Obtiene el HOME de SAAI
Declare Function saai_path Lib "sqlclin" (ByVal Socket As Long, _
ByVal PATH As String) As Long

' Cierra el Socket
Declare Function cierra Lib "sqlclin" (ByVal Socket As Long) As Long

Declare Function vbstrstok Lib "sqlclin" (ByVal s1 As String, _
ByVal s2 As String, ByVal tok As String, ByVal primero As Long) As Long

'Función para obtener el HOME de SAAI
Public Function ObtienePATH(Hostname As String, Servicio As String, strPATH As String) As
Long
Dim Socket As Long, PATH As String
ObtienePATH = conecta(Hostname, Servicio, Socket)
If ObtienePATH <> 0 Then
    Err.Number = ObtienePATH
    Err.Description = "[ObtienePATH] Error al conectarse con el servidor"

```

```

        Err.Source = Err.Source & "[SQLCLIN][CONECTA]"
    Exit Function
End If
PATH = String(80, 0)
ObtienePATH = saai_path(Socket, PATH)
If (ObtienePATH <> 0) Then
    Err.Number = ObtienePATH
    Err.Description = "[ObtienePATH] Error al obtener el path"
    Err.Source = Err.Source & "[SQLCLIN][SAAI_PATH]"
Else
    strPATH = QuitaChr(PATH, vbNullChar)
End If
cierra Socket
End Function

```

Fin Sqlclin.bas

También fue necesario construir una biblioteca dinámica (dll) en lenguaje C a ser empleada por el cliente desarrollado en Visual Basic para conectarse a la base de datos y obtener la información de la misma. A continuación se muestra un extracto del código.

Sqlclin.c:

```

/****  DLL para ejecucion de comandos SQL          ****

#include <windows.h>
#include <winuser.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <winsock.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <string.h>
#include <errno.h>
#include <io.h>
#include <fcntl.h>

#define FAR _stdcall
#define BUFFSIZE      1024
#define servicioqry   "qrysrv"

// tipo de codigo devuelto por la consulta
#define QRY_EOD        "ATMTA0000"
#define QRY_SAAIPATH  "ATMTA0101"
#define QRY_DATOS     "ATMTA0110"
#define QRY_CODIGO    "ATMTA1001"
#define QRY_OPENDB    "ATMTA1111"
#define QRY_ARCHIVO   "ATMTA1000"

// Tipo de operación a ejecutarse
#define SQLEXEC        0
#define SQLDYNA        1

// dirección de avance del cursor abierto para consulta
#define SQLPRIMERO     0
#define SQLULTIMO      1
#define SQLSIGUIENTE   2
#define SQLANTERIOR    3
#define SQLCLOSE       4
#define SQLBEGINTRANS  5
#define SQLCOMMIT      6
#define SQLROLLBACK    7
#define SQLABREBD      8
#define SQLFINDFIRST   9
#define SQLFINDNEXT    10
#define SQLFINDLAST    11

```

```

// Función interna para el envío de la petición y recuperación de los datos
int EnviaPeticiónYRecibeDatos(int fd_socket, char *buffer, char *cadena)
{
    int errnum;

    auxlong = strlen(buffer);
    longmsg = htonl(auxlong);

    // Se realiza el envío de la petición y recuperación de los datos
    if( !( ( errnum=30, (( send(fd_socket, (char *)&longmsg, sizeof(u_long),0))
        != SOCKET_ERROR) //longitud
        && ( ++errnum, (( send(fd_socket, buffer, auxlong, 0))
        != SOCKET_ERROR) //qry
        && ( ++errnum, (( recv(fd_socket, (char *)&longmsg, sizeof(u_long),0))
        != SOCKET_ERROR) //longitud
        && ( ++errnum, (( cadena = (char *)calloc(htonl(longmsg), 1)) ))
        && ( ++errnum, (( auxlong = recv(fd_socket, cadena, htonl(longmsg), 0))
        != SOCKET_ERROR) //datos
        && ( ++errnum, ( auxlong == htonl(longmsg) ) )
    ))
        return(errnum);

    return(0);
}

// Función para la ejecución de la sentencia SQL
int FAR ejecuta_sql(int FAR fd_socket, int FAR operacion, BSTR qry,
    BSTR titulos, BSTR tipos, BSTR tamaño,
    BSTR datos, int FAR *numregs)
{
    int errnum=0;
    u_long longmsg, auxlong;
    char buffer[BUFFSIZE];
    char *cadena;
    char noregs[7] = "999999";

    // No se especificó sentencia SQL a ejecutar
    if( !(auxlong = strlen((LPSTR)qry)) )
        return(1);

    // Se inicializa el espacio de memoria
    (void)memset( buffer, 0, 1024);

    // se reciben titulos de campo o codigo de error
    if( (errnum = EnviaPeticiónYRecibeDatos(fd_socket, (LPSTR)qry, cadena)) )
    {
        (void)closesocket(fd_socket);
        return(errnum);
    }

    // Se ejecutó un comando SQL y se recibió un código de error
    if ( operacion == SQLEXP || strstr(cadena, QRY_CODIGO) )
    {
        (void)closesocket(fd_socket);
        (void)strcpy((LPSTR)datos, cadena + 15);
        return( atoi(cadena + 9) );
    }

    // Se arman las cadenas que conforman el registro
    (void)strcpy((LPSTR)titulos, strstr(cadena, QRY_EOD));
    (void)strcpy((LPSTR)tipos, strstr(NULL, QRY_EOD));
    (void)strcpy((LPSTR)tamaño, strstr(NULL, QRY_EOD));
    (void)strcpy(buffer, strstr(NULL, QRY_EOD));
    (void)strncpy(noregs, buffer, 6);
    *numregs = atoi(noregs);
    (void)strcpy((LPSTR)datos, buffer + 6);
    return(0);
}

```

Capítulo 5. Implantación

Capítulo 1. Implantación

Esta fase tiene como propósito asegurar que el sistema desarrollado se encuentre disponible para los usuarios finales, que puede incluir algunas pruebas previas a la liberación y realizar ajustes mínimos basados en la retroalimentación de los usuarios.

5.1 Pruebas de aceptación

Como se mencionó antes, son las pruebas realizadas previo a distribuir el software. La finalidad es verificar que el software está listo y que puede ser usado por los usuarios para realizar las tareas para lo cual fue desarrollado.

En este caso se aplicarán los casos de prueba definidos en la fase de diseño por el área de desarrollo más aquellas que el equipo de pruebas y los usuarios hayan definido.

Existen tres estrategias para realizar las pruebas de aceptación

- Aceptación formal
- Aceptación informal o pruebas alpha
- Pruebas beta

Aceptación formal

Estas pruebas normalmente son consideradas como una extensión de las pruebas de sistema. Son planeadas y diseñadas tan cuidadosa, y tan detalladamente como las de sistema. Los casos de prueba seleccionados deben ser un subconjunto de los ejecutados en las pruebas de sistema.

Aceptación informal

Éstas no son tan rigurosas ni tan detalladas como las anteriores. Las funciones y tareas de negocio a probarse y documentarse no tiene un caso específico a ejecutar. Cada probador determina que hacer. Esta estrategia no es tan controlada como la formal y es más subjetiva. Normalmente es realizada por los usuarios finales de la organización.

Pruebas Beta

Las pruebas beta son las menos controladas de las tres estrategias. El nivel de detalle, los datos y como llevarlas depende de cada probador. Cada uno de ellos es responsable de crear su ambiente, seleccionar sus datos, qué características y funcionalidad probar. Además de establecer los criterios de aceptación.

Las pruebas beta son realizadas por los usuarios finales y es la más subjetiva de las estrategias de aceptación.

Para el caso de la organización, el usuario de negocio o informático es quién se encarga de definir y preparar las pruebas de aceptación formales considerando el alto riesgo e impacto que las aplicaciones pueden tener para las operaciones de comercio exterior del país.

5.2 Capacitación de los usuarios de la aplicación

Tiene como propósito, garantizar que los usuarios y operadores, hayan obtenido los elementos necesarios para comenzar la operación del sistema.

Se realiza esta actividad de acuerdo con el plan de capacitación definido en la fase de diseño.

5.3 Liberación y distribución de la aplicación

Se establece como será distribuida y puesta en operación, además se establecen de manera clara los componentes finales del sistema que se implantarán.

A continuación se muestran como ejemplo dos matrices que contienen los fuentes que conforman la aplicación (figura 5.1.1) y cómo deben ser integrados para su liberación (figura 5.1.2):

CONTROL DE CAMBIOS					
PROYECTO/SUBPROYECTO	REFIS/VALIDA				
INSTALABLES AFECTADOS	valgral				
CUENTA DE TRABAJO DEL USUARIO	atejeda	NOMBRE	Alejandro F. Tejada Muñoz	SUPERVISOR	
DIRECTORIO PARA EDICIÓN/INGRESO	\$HOME/REFIS/VALIDA				
FECHA DE EXTRACCIÓN PARA EDICIÓN	-			FECHA DE INGRESO DE CAMBIOS	10/10/03
JUSTIFICACIÓN(ES) DE LOS CAMBIOS	Proyecto Control de Recintos Fiscalizados				
ARCHIVO A MODIFICAR	VERSION		DESCRIPCIÓN DE LOS CAMBIOS		
	ANTERIOR	NUEVA			
valgral.c		1.0	Validador genérico.		
valgral.h		1.0	Declaración de funciones.		
funcs.ec		1.0	Definición de funciones de acceso a base de datos.		
funcs.h		1.0	Declaración de funciones.		
makefile		1.0	Script de compilación		

figura 5.1.1 Hoja de control de cambios del sistema.

MATRIZ DE ACTUALIZACIÓN									
INSTALABLE	TIPO	VERSIÓN.	EQUIPO	DIRECTORIO DESTINO	PERM.	PROP.	GRUPO	DESCRIPCIÓN	OBSERVACIONES
valgral	X	1.0	A	/opt/refis/bin	100	refis	refis	Validador genérico	Programación nueva
genactas	X	1.0	A	/opt/refis/bin	100	refis	refis	Generación automática de actas de abandono	Programación nueva
sgencdec	X	1.0	A	/opt/refis/bin	100	refis	refis	Generación y apertura de sobres digitales	Programación nueva
sqlsrvr	X	1.0	A	/opt/refis/bin	100	refis	refis	Servicio de comandos SQL remotos	Programación nueva

Tipo: [X] Ejecutable [F] Forma
 [O] Código objeto [A] Archivo ascii
 [B] Archivo binario [AL] Biblioteca de archivo
 [SL] Biblioteca compartida [AZ] Archivo ascii comprimido
 [BZ] Archivo binario comprimido [SS] Script shell, awk, etc

Equipos: [A] Aduana UNIZ [P] Aduana PC's (*)

*.- En estos casos especificar a detalle el equipo en el que se hará la instalación.

figura 5.1.2 Matriz de actualización del sistema.

La figura 5.1.3 muestra un diagrama de distribución con los componentes del sistema desarrollado, así como los equipos en los cuales serán instalados cada uno de ellos.

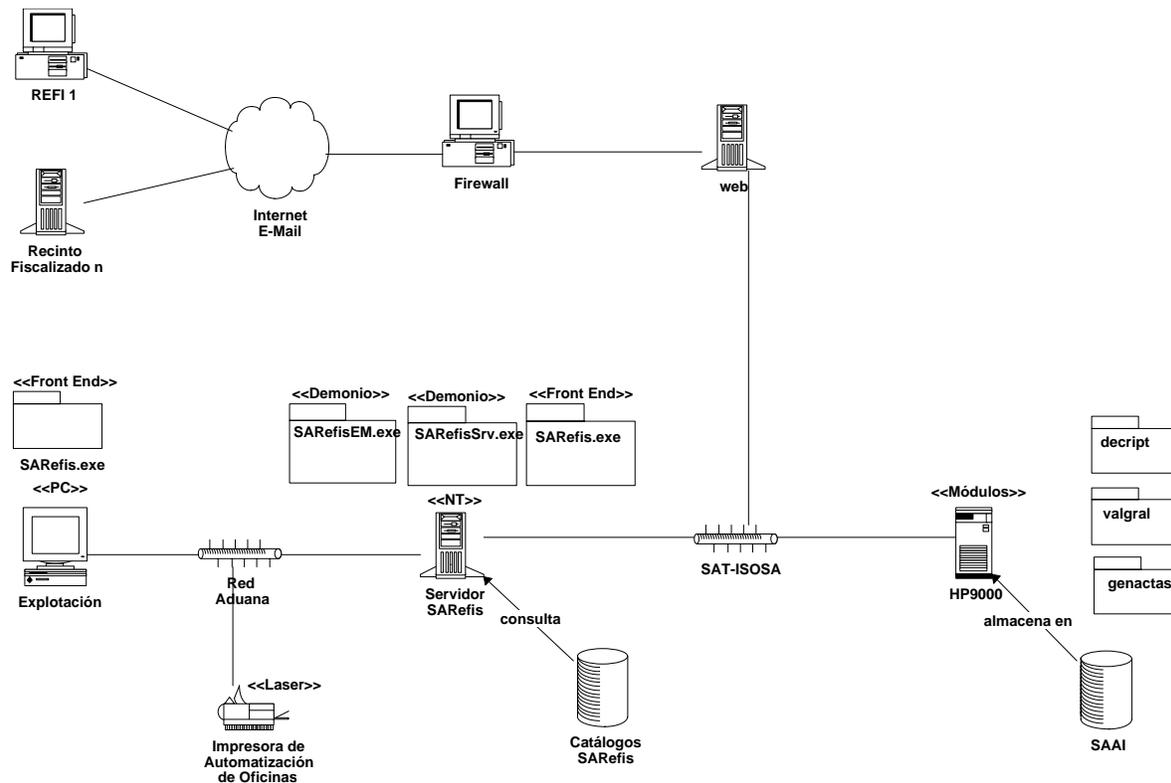


figura 5.1.3 Diagrama de distribución de la aplicación

Conclusiones

Conclusiones

Con la realización del sistema se logró cumplir con el objetivo planteado al poder controlar a los recintos fiscalizados apegados a la normatividad en cuanto a la forma y en los tiempos establecidos por ella.

Se lograron los siguientes beneficios:

- Eliminar los recursos asignados por la aduana para la captación y transformación de la información recibida a los formatos de control empleados en cada una de ellos.
- Seguridad en el almacenamiento de la información al contarse con medios permanentes dentro del servidor. Así mismo, al ser un proceso automatizado y no haber intervención humana, la información no puede ser alterada.
- Seguridad en el intercambio de la información al emplearse técnicas criptográficas para codificar los archivos, además de garantizar el origen de la misma.
- Abatir el rezago existente en el seguimiento de los abandonos al controlar el sistema de manera automática las mercancías que han causado abandono.

Del empleo de la metodología y modelo definido por la institución para el desarrollo de sistemas, se detectaron los siguientes detalles:

- Aunque se consiguieron los objetivos para los cuales fue desarrollado el sistema, el empleo de una metodología de desarrollo estructurado no es la más adecuada para el desarrollo de una aplicación cliente/servidor donde el front-end fue construido con una herramienta de desarrollo orientada a eventos, ya que los modelos de análisis y diseño estructurado no sirven para ello y resulta bastante complicado tratar ajustar esta solución a ellos, para ello se deben emplear los modelos orientados a objetos.
- El emplear el modelo en cascada sin realizar iteraciones dentro de cada una de las fases, como se realiza dentro de la institución, tiene los siguientes inconvenientes:

La incorporación de cambios en los requerimientos repercute negativamente en el cumplimiento de los plazos establecidos ya que al no existir las iteraciones por fase, no es posible planear una siguiente iteración para considerarlo, a menos que sea al iniciar un nuevo proceso o se incorpore sobre el actual con el consiguiente impacto en los tiempos

previstos originalmente, además de no existir una disciplina para la administración de los requerimientos.

Cumplir con el concepto de escalabilidad a la larga resulta más difícil de esta manera, pues se debe tener una visión muy clara desde un inicio del crecimiento esperado, o considerar emplear una arquitectura en capas para reducir al mínimo los impactos generados al incorporar nuevos componentes.

Se tiene un alto costo de pruebas y corrección de defectos ya que las pruebas son realizadas al final del proceso, si consideramos el costo de corregir errores en las fases tempranas del desarrollo es de una a diez veces, y en etapas tardías puede resultar de cien a doscientas veces mayor, además de que muchos de los riesgos involucrados alrededor del proyecto no son detectados sino hasta el momento de realizar la integración de los componentes.

En pocas palabras, con el modelo en cascada aplicado de esta manera, todo parece estar bien hasta que se acerca el fin del proyecto y en ocasiones (el mejor de los casos) durante la integración de los módulos. Por lo anterior sería recomendable para la organización modificar el modelo actual y pensar en emplear un proceso iterativo con una adecuada administración de requerimientos, además de la utilización de técnicas de análisis y diseño orientado a objetos.

Bibliografía

Kendall, Kenneth E.

Kendall, Julie E.

Systems Analysis and Design, Third Edition

Prentice Hall, 1995

Yourdon, Edward

Modern Structured Analysis

Prentice Hall, 1989

Pressman, Roger S.

Software Engineering, A Practitioner's Approach, Fourth Edition

Mc Graw Hill, 1997

Rubi, Tony

User Interface Design for Computer Systems

Halstead Press, 1988.

Mountford, S. Joy

The Art of Human-Computer Interface Design

Addison-Wesley, 1990

Ruble, David A.

Análisis y Diseño Práctico para Sistemas Cliente Servidor con GUI.

Prentice Hall Hispanoamericana, S. A., 1998

Meyers, Gelnfor J.

El Arte de Probar el Software

El Ateneo, 1984

Ullman Jeffrey D. y Widom, Jeniffer

Introducción a los Sistemas de Bases de Datos

Prentice Hall, México 1999

Kaliski Jr., Burton S.

An Overview of the PKCS Standards

An RSA Laboratories Technical Note, Revised November 1, 1993.

Rational Unified Process version.3003.06.00.65

Rational Software Corporation

Cortina Velarde, Francisco José

Ley Aduanera (Análisis y Comentarios)

Editorial Themis