

34



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGÓN"**

**DESARROLLO DEL SISTEMA FIXBAN
CLIENTE BANXICO PARA EL CALCULO DE
LA EQUIVALENCIA DEL PESO CONTRA EL
DOLAR**

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A N :
ILIANA MENESES HERNÁNDEZ
MARÍA BELEN BENÍTEZ QUIJANO

ASESOR: ING. DONACIANO JIMENEZ VAZQUEZ

SAN JUAN DE ARAGÓN, EDO.MÉXICO.

2002

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACION DISCONTINUA

Dedicatoria

*"Los sueños se pueden hacer realidad,
sólo hay que pagar el precio"
Belén Benítez*

A Dios que me permite vivir y me enseña día a día que hermoso es sentir el amor en cada persona, en cada momento en cada objetivo que realizo por este camino. Por poner en mi camino a tantas personas maravillosas.

A mis padres, personas tan invaluable, quienes me dieron la vida y sobre todo el amor, confianza y fé. Nunca terminaré de pagar y agradecer todo lo que son y lo que han hecho por mí, ésta solo es una forma de hacerlo.

A mi madre, por ser fuente de inspiración, sensibilidad y fortaleza.

A mi padre, por sembrar en mí la superación, respeto y valores humanos.

A mis hermanas, hermosas flores, quienes me han apoyado y enseñado con sus propias vidas; las admiro realmente y las amo.

A mi esposo, por su apoyo e insistencia para que lograra este objetivo más.

A los profesores de la Escuela Nacional de Estudios Profesionales Aragón, por los conocimientos compartidos.

A la gloriosa Universidad Nacional Autónoma de México quien me cogió en su seno y me permite liberar este espíritu de superación intelectual.

A Iliana Meneses, amiga y casi mi hermana, por sus consejos y ejemplo.

Al Ing. Donaciano Jiménez quien ha contribuido en gran parte a terminar este proyecto. Por su apoyo a muchos estudiantes como nosotros que buscamos realizar este gran sueño.

A mis amigos y compañeros de trabajo, quienes me escuchan y con sus comentarios y cariño me hacen ser más fuerte.

A mis líderes laborales quienes me enseñan con su ejemplo que los sueños se pueden realizar.

A Jesús Gamboa quien desde el cielo me cuida y me protege.

A los jurados por sus valiosos conocimientos y comentarios.

Y a todas aquellas personas que de una u otra forma, colaboraron o participaron en la realización de este proyecto, hago extensivo mi más sincero agradecimiento. A todos ustedes gracias por existir en este tiempo y espacio.

TESIS CON
FALLA DE ORIGEN

Dedicatoria

A Lidia Hernández Quintero,
Cuyo principal interés fue siempre mi formación académica y mi superación profesional.
Quien nunca escatimó por ofrecerme un mundo de oportunidades.
Quien con su amor, apoyo y ejemplo me ayudó a lograrlo. Gracias madre.

A Nick Cardo mi esposo,
Por nunca dudar y siempre dar.
Porque mis retos y mis logros son los suyos.
Por ayudarme a enriquecer a nuestra familia.
Gracias por amar.

Iliana Meneses Hernández.

TESIS CON
FALLA DE ORIGEN

CONTENIDO

Prefacio iv

Lista de tablas y figuras viii

Capítulo 1. Introducción 1

1.1 Ubicación del proyecto 1

1.2 Objetivo 1

1.3 Marco teórico 1

1.3.1 Ciclo de vida del modelo estructurado 2

1.3.2 Herramientas de modelado y planeación 4

1.3.2.1 Red de tareas 5

1.3.2.2 Diagrama de Gantt 6

1.3.2.3 Diagrama de flujo de datos 7

1.3.2.4 Diccionario de datos 9

1.3.2.5 Miniespecificaciones 9

1.3.2.6 Diagrama de estructuras 10

1.3.2.7 Interfaz gráfica del usuario (GUI) 12

1.3.2.8 Enfoque orientado a objetos 13

1.3.3 Metodología de análisis estructurado 15

1.3.3.1 El modelo ambiental 15

1.3.3.2 El modelo funcional 17

1.3.4 Metodología de diseño estructurado 17

1.3.4.1 El modelo de implementación del usuario 18

1.3.4.2 El modelo de implementación del sistema 18

1.3.4.3 El modelo de implementación del programa 18

Capítulo 2. Análisis 19

2.1 Estudio preliminar 19

2.1.1 Juntas de trabajo 19

2.1.2 Especificación de requerimientos 24

2.1.3 Planeación 30

TESIS CON
FALLA DE ORIGEN

- 2.2 La especificación estructurada 34
- 2.2.1 El modelo esencial del sistema FixBan cliente Banxico 34
- 2.2.2 El modelo ambiental del sistema FixBan cliente Banxico 34
- 2.2.3 El modelo funcional del sistema FixBan cliente Banxico 37

Capítulo 3. Diseño 61

- 3.1 El modelo de implementación del usuario para FixBan cliente Banxico 61
 - 3.1.1 Límites de automatización 61
 - 3.1.2 Formatos de entrada y salida 69
 - 3.1.3 Diseño de formas 75
 - 3.1.4 Límites operacionales 83
 - 3.1.4.1 Volumen de datos 83
 - 3.1.4.2 Tiempo de respuesta 85
 - 3.1.4.3 Limitaciones políticas 85
 - 3.1.4.4 Seguridad 85
 - 3.1.4.5 Disponibilidad 85
- 3.2 El modelo de implementación del sistema para FixBan cliente Banxico 86
 - 3.2.1 El modelo de procesadores 86
 - 3.2.2 El modelo de tareas 87
- 3.3 El modelo de implementación del programa para FixBan cliente Banxico 87

Capítulo 4. Implementación 96

- 4.1 La tarea de comunicaciones 96
- 4.2 La tarea de atención al FEC 98
- 4.3 La tarea de atención al operador 103
 - 4.3.1 La pantalla de entrada 103
 - 4.3.2 La pantalla principal 105
 - 4.3.3 La pantalla de configuración 107
 - 4.3.4 La pantalla de ejecución de las subastas 110
 - 4.3.5 La pantalla de históricos 113
 - 4.3.6 La pantalla de cambio de password 115
 - 4.3.7 La pantalla de distribución de indicaciones 116
 - 4.3.8 La pantalla de reportes y cartas 118

4.3.9	La pantalla de impresión de asignaciones	121
4.3.10	La pantalla de selección del mejor arbitraje	124
4.3.11	La pantalla de desconexión	125
4.3.12	La pantalla de mensajes	126
4.4	Módulos de clase	128
4.5	Pruebas	137

Capítulo 5. Conclusiones 140

Bibliografía 152

Apéndice A. Diccionario de datos 153

TESIS CON
FALLA DE ORIGEN

Prefacio

En este trabajo de tesis pretendemos describir las fases de desarrollo del sistema FixBan Cliente Banxico empleado por el Banco de México. Esta herramienta de software se ocupa del control de las operaciones para determinar la equivalencia del peso mexicano contra el dólar estadounidense.

El documento está organizado en cinco capítulos: Introducción, Análisis, Diseño, Implementación y Conclusiones.

En el primer capítulo se describen los objetivos del proyecto y las herramientas teóricas en las cuales se sustenta el desarrollo del mismo. El capítulo dos muestra las actividades realizadas para la elaboración de un modelo para el análisis de los requerimientos del usuario. Dicho modelo sirvió como base para diseñar en el capítulo tres, los alcances y límites del sistema, las fronteras con el usuario y el resto del medio ambiente, y la organización de las partes del mismo, enfocados en proveer una mejor herramienta para satisfacer las necesidades y lograr los objetivos del usuario/cliente.

En el capítulo cuatro se muestran los módulos del sistema a través de las pantallas resultantes. Se incluyen funciones que son parte del programa hecho en Visual Basic, sin embargo, con el fin de mantener este documento conciso pero descriptivo, se optó solo por incluir sólo algunas de ellas. El capítulo cuatro muestra también, las pruebas realizadas para evaluar el desempeño del sistema.

Para finalizar en el capítulo cinco se establecen las conclusiones como resultado de las actividades y la experiencia adquirida en cada fase. Se hacen las recomendaciones que se consideran oportunas para el mantenimiento y mejora del sistema.

Antecedentes y descripción del problema

El Banco de México¹ es la institución que regula las operaciones para determinar el tipo de cambio de la moneda nacional con respecto a la moneda extranjera dentro de la República Mexicana.

Para determinar la equivalencia del peso mexicano con respecto al dólar estadounidense, y de acuerdo con el Diario Oficial de la Federación [Diario Oficial 1996], el tipo de cambio se realiza como sigue:

Cada día hábil bancario se obtienen cotizaciones del tipo de cambio de compra y venta del dólar de instituciones de crédito cuyas operaciones reflejen las condiciones predominantes en el mercado de cambios al mayoreo para operaciones liquidables el segundo día hábil bancario siguiente a la fecha de tal cotización. Las cotizaciones se solicitan en tres lapsos: de las 9:00 a las 9:59 horas, de las 10:00 a las 10:59 horas y de las 11:00 a las 12:00 horas. Cada institución financiera (Banamex, Citibank, BBVA-Bancomer, Banorte, Bital, etc.) sólo puede efectuar cotizaciones en un lapso por día.

¹ Banco de México, <http://www.banxico.org.mx/>

Para tal efecto, se realiza aleatoriamente dentro de cada una de dichas etapas una subasta de quince minutos para solicitar las cotizaciones mencionadas a cuatro instituciones de crédito, las cuales son seleccionadas por la Subgerencia de Cambios Nacionales del Banco de México. Las cotizaciones se solicitan por un monto que refleje la práctica predominante en el mercado de cambios al mayoreo. El monto es el mismo para los tres lapsos mencionados.

El tipo de cambio de equilibrio se calcula aplicando un algoritmo matemático a las cotizaciones finales de cada etapa, posteriormente se obtiene el promedio aritmético de los tres tipos de cambio subastados cerrando el resultado a cuatro decimales, a este resultado se le conoce como FIX o FIX del día.

Las cotizaciones y el resultado del FIX son confirmados el mismo día a la Subgerencia de Cambios Nacionales mediante un escrito u otro medio que deje constancia de la confirmación.

El Banco de México publica el tipo de cambio resultante (FIX), en el Diario Oficial de la Federación y otros medios como Internet, el día hábil bancario inmediato siguiente.

Mientras la Subgerencia de Cambios Nacionales se encarga de la operación del FIX, la Subgerencia de Infraestructura Informática está a cargo de su mantenimiento. De 1996 a 1999 se contó con un sistema automático, conocido como *FixBan 1.3*, y un sistema para impresión de reportes. Conforme nuevas características eran requeridas se fueron desarrollando sistemas por separado, obteniendo como resultado un sistema principal y varias utilerías. Por otro lado, dentro de la misma Subgerencia de Infraestructura Informática, se desarrolló un protocolo de comunicación conocido como FEC (Front End de Comunicaciones) [Hernández 1999], que resultó ser conveniente en sistemas cliente/servidor para mantener la seguridad de los servidores y para mejorar la velocidad de comunicación.

La Subgerencia de Infraestructura Informática decidió que los servicios que estuvieran a su cargo deberían ser actualizados para soportar el protocolo de comunicación del FEC. De tal manera que se requirió de un sistema que sustituyera al *FixBan 1.3*, el cual debería integrar la funcionalidad de cada utilería e incorporar el protocolo de comunicación anteriormente mencionado.

Descripción breve del proyecto

En esta sección pretendemos describir brevemente las partes que conforman el sistema que reemplazará al *FixBan 1.3*, al cual se le ha llamado *FixBan 2.0*.

Como el *FixBan 2.0* fue desarrollado por un equipo de trabajo, el objetivo de esta sección es describir cuál es la parte que nos correspondió desarrollar y que estamos presentando en este proyecto de tesis.

La figura 1.1 muestra las partes que componen al *FixBan 2.0*, las cuales son las siguientes:

- Servidor *FixBan 2.0*.
- FEC de comunicaciones.
- Clientes.

TESIS CON
FALLA DE ORIGEN

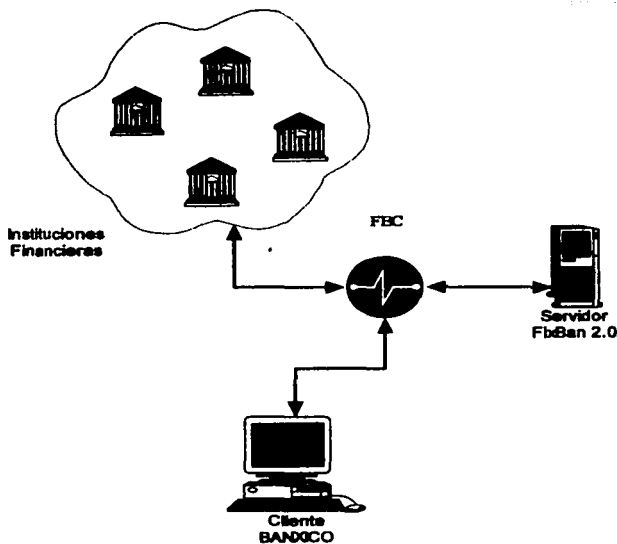


Figura 1 Sistema FixBan 2.0.

El *servidor*, su función principal es aplicar el algoritmo para el cálculo del FIX. Registra los sucesos ocurridos en todas las sesiones del día, de tal manera que mantiene una bitácora interna para que el cliente Banxico siempre cuente con la información veraz más reciente y con la cual se llevan a cabo las subastas.

El *FEC de comunicaciones*, es un sistema que sirve como interfaz para el intercambio de mensajes entre clientes y servidores. Define el protocolo utilizado para este fin. Autentifica a clientes y a servidores, y lleva una bitácora del estado de conexión o desconexión de ambos. Distribuye los mensajes a los clientes y servidores adecuados, de tal manera que los clientes no conocen la localización de los servidores y viceversa.

Los *clientes* son los sistemas que sirven como interfaz entre los usuarios y el sistema FixBan 2.0. Existen dos tipos de clientes:

- El cliente Banxico, es el sistema operado por el Banco de México, desde el cual se controla la realización de las subastas para el cálculo del FIX.

- Los clientes de las instituciones financieras, son los sistemas que permiten a los bancos conectarse al FixBan y participar en las subastas destinadas al cálculo del FIX.

El FixBan 2.0 cliente Banxico es el sistema presentado en éste proyecto de tesis.

TESIS CON
FALLA DE ORIGEN

Lista de tablas y figuras

Figura	Nombre	Pág.
Figura 1	Sistema FixBan 2.0.	vi
Figura 1.1	Ciclo de vida del modelo estructurado.	2
Figura 1.2	Componentes de una tarea.	5
Figura 1.3	Red de tareas.	6
Figura 1.4	Diagrama de Gantt.	7
Figura 1.5	Diagrama de flujo de Datos.	8
Figura 1.6.	Diagrama de estructuras.	11
Figura 1.7	Vocabulario canónico.	12
Figura 1.8	Ejemplo de una clase.	14
Figura 1.9	Instancia de una clase.	14
Figura 2.1	Red de tareas del sistema FixBan cliente Banxico.	31
Figura 2.2	Diagrama de Gantt del sistema FixBan cliente Banxico.	32-33
Figura 2.3	Fronteras del sistema.	34
Figura 2.4	Diagrama de contexto.	35
Figura 2.5	DFDs de nivel 1.	37-41
Figura 2.6	DFDs de bajo nivel.	41-46
Figura 2.7	Diagrama de transición de estados.	59
Figura 3.1	Límites de automatización.	62-68
Figura 3.2	Formatos de salida.	69-74
Figura 3.3	Diseño del GUI.	75-82

Figura 3.4	Modelo de procesadores.	86
Figura 3.5	Diagrama de estructuras.	88-95
Figura 4.1	La pantalla de entrada.	103
Figura 4.2	La pantalla principal.	105
Figura 4.3	La pantalla de la configuración.	108
Figura 4.4	La pantalla de las subastas.	110
Figura 4.5	La pantalla del histórico.	113
Figura 4.6	La pantalla de cambio de password.	115
Figura 4.7	La pantalla de distribución de indicaciones.	117
Figura 4.8	La pantalla de generación de cartas.	119-120
Figura 4.9	La pantalla de impresión de asignaciones.	121
Figura 4.10	La pantalla de selección del mejor arbitraje.	125
Figura 4.11	La pantalla de información de mensajes.	127
Figura 4.12	Selección de participantes.	137
Figura 4.13	Configuración de la subasta.	138
Figura 4.14	Ejecución de la subasta.	139
Figura 4.15	Estado en espera de clientes.	140
Figura 4.16	Fin de la sesión.	141
Figura 4.17	Histórico de la primera sesión del día.	142
Figura 4.18	Inicio de la segunda subasta.	143
Figura 4.19	Arbitrajes de la segunda sesión.	144
Figura 4.20	Selección del mejor arbitraje.	145
Figura 4.21	Resultados del FIX y TCVM del día.	146

Figura 4.22	Carta para Comunicación Social.	147
Figura 4.23	Publicación de los resultados en Internet.	147
Figura 4.24	Asignaciones del día.	148
Tabla 1.1.	Notación empleada en un diccionario de datos.	9
Tabla 2.1	Lista de Eventos.	36
Tabla 3.1	Formatos del protocolo del FEC usados por el cliente Banxico.	83-84
Tabla 3.2	Modelo de tareas.	87

TESIS CON
FALLA DE ORIGEN

Capítulo 1

INTRODUCCIÓN

Este proyecto de tesis pretende describir las actividades realizadas durante la elaboración del sistema FixBan 2.0 cliente Banxico, utilizado por el Banco de México para calcular el valor de equilibrio de la moneda mexicana contra el dólar estadounidense o FIX.

En este primer capítulo se abordarán los antecedentes que nos expliquen por que se llegó a su desarrollo, y se describirán los fundamentos teóricos y de computación en los que está sustentado para optimizar este importante proceso financiero.

1.1 Ubicación del proyecto

El área de la investigación en la que se ubica éste proyecto es la Ingeniería de Sistemas. Wymore [Wymore 1993] define la Ingeniería de Sistemas como *"la disciplina intelectual, académica y profesional que tiene como interés principal la responsabilidad de asegurar que todas las requerimientos de un sistema de software / hardware / bioaware* sean satisfechos a través del ciclo de vida de dicho sistema."* Particularmente se desarrollará un producto de Ingeniería del Software.

1.2 Objetivo

Se desarrollará el cliente Banxico para el sistema FixBan 2.0, desde el cual se controlan los procesos para calcular la equivalencia del peso mexicano contra el dólar estadounidense o FIX.

Esta pieza de software usará como base el FEC (Front End de Comunicaciones) para su comunicación con el resto del sistema.

Con el uso de conocimientos adquiridos en la carrera de Ingeniería en Computación y en el ámbito laboral: Planeación de Sistemas e Ingeniería del Software (herramientas de quinta generación, análisis y diseño de sistemas, programación), se pretende que el cliente Banxico para el FixBan 2.0, brinde a la Subgerencia de Mercados Nacionales del Banco de México, un mejor medio para controlar los procesos del FIX, más rápido, eficiente y confiable.

1.3 Marco teórico

El desarrollo de sistemas de software se ve respaldado ampliamente por diversas herramientas teóricas y prácticas.

¹ Banco de México, <http://www.banxico.org.mx/>

Teóricamente el desarrollo de un software sigue un ciclo de vida, prácticamente se han desarrollado metodologías que sugieren el uso de herramientas que puedan ser ajustadas al ciclo de vida de un sistema en particular, el utilizado por nosotros es el modelo estructurado que propone Edward Yourdon quien nos dice: "Cuando un profesionalista se enfrenta al campo de trabajo, tiene pocas herramientas al elaborar un proyecto de desarrollo de sistemas, sin embargo, este modelo es uno de los más sencillos y efectivos para encausarse como un verdadero profesionalista en el desarrollo de sistemas".

1.3.1 Ciclo de vida del modelo estructurado

El modelo estructurado se compone por nueve fases y tres terminadores [Yourdon, 1989] como se muestra en la figura 1.2. Los terminadores son los usuarios, los administradores, y el personal de operaciones, los cuales son individuos o grupos que proporcionan las entradas al sistema, y son los beneficiados finales.

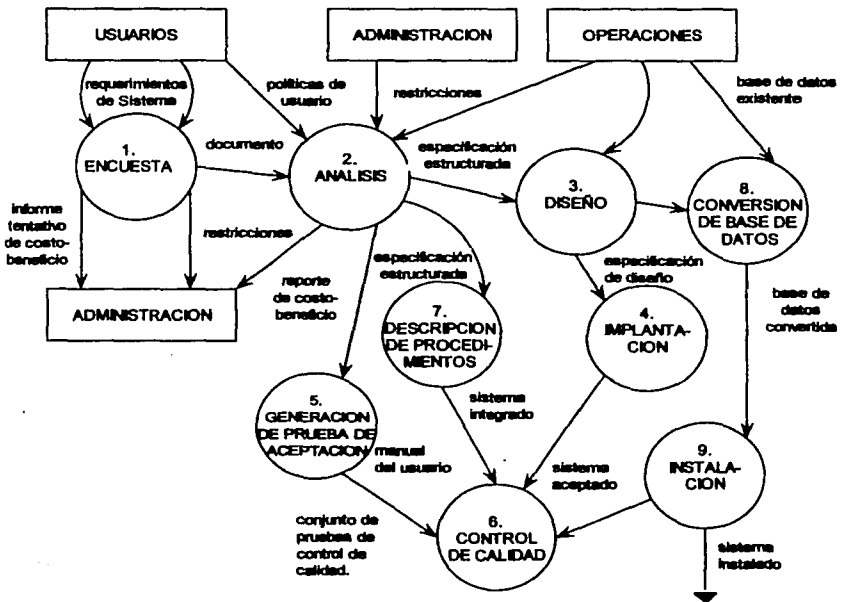


Figura 1.1 Ciclo de vida del modelo estructurado.

TESIS CON FALLA DE ORIGEN

Fase 1: La encuesta

Esta actividad también se conoce como el estudio de factibilidad o como el estudio inicial de negocios. Empieza cuando el usuario solicita que una o más partes de su sistema se automatizen. Los principales objetivos de la encuesta son:

- Identificar a los usuarios principales y crear "un campo de actividad" inicial para el sistema.
- Identificar las deficiencias en el ambiente actual del usuario.
- Establecer metas y objetivos para un sistema nuevo.
- Determinar si es factible automatizar el sistema y de ser así, sugerir escenarios aceptables.
- Generar la especificación de los requerimientos del usuario.
- Preparar el esquema que se usará para guiar el resto del proyecto.

Usualmente la encuesta ocupa sólo del cinco al diez por ciento del tiempo y de los recursos de todo el proyecto, es sabido que para proyectos pequeños pudiera ni siquiera ser una actividad formal. A pesar de todo lo anterior, es una actividad importante debido a que la administración pudiera decidir cancelar el proyecto si no parece atractivo desde el punto de vista de costo-beneficio.

Fase 2: Análisis

El propósito principal del análisis es transformar sus dos entradas principales: los requerimientos del usuario y el esquema del proyecto en una Especificación Estructurada.

La especificación estructurada consiste en modelar el ambiente del usuario a través del uso de diagramas y herramientas, tales como: diagramas de flujo de datos, diagramas de transición de estado, diccionario de datos, miniespecificaciones, etc.

El proceso paso a paso del análisis estructurado implica el desarrollo de un Modelo Ambiental y el desarrollo de un Modelo de Comportamiento, los cuales se combinan para formar el Modelo Esencial que representa una descripción formal de lo que el nuevo sistema debe hacer, independientemente de la naturaleza de la tecnología que se use para cubrir los requerimientos.

Fase 3: Diseño

El diseño se dedica a asignar porciones de la especificación estructurada (modelo esencial) a procesadores adecuados (máquinas o humanos) y a labores apropiadas (tareas, particiones, etc.) dentro de cada procesador. Para cada tarea en particular, la actividad de diseño se enfoca en la creación de una jerarquía apropiada de módulos de programas y de interfases entre ellos para implantar la especificación creada en la actividad del análisis.

Fase 4: Implementación

Esta actividad incluye la codificación e integración de los módulos en un esqueleto progresivamente más complejo del sistema final. Por eso, esta actividad incluye tanto programación estructurada como implantación descendente.

Fase 5: Generación de pruebas de aceptación

La especificación estructurada debe contener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario. Por eso, una vez generada la especificación, se puede comenzar a producir un conjunto de casos de prueba de aceptación.

Fase 6: Garantía de calidad

La garantía de calidad también se conoce como la prueba final. Esta actividad requiere como entradas los datos de la prueba de aceptación generados en la actividad previa y el sistema integrado producido en la implantación.

Fase 7: Elaboración de procedimientos

Esta actividad implica la descripción formal de las partes del sistema que se harán en forma manual, lo mismo que la descripción de como interactuarán los usuarios con la parte automatizada. El producto de esta actividad es el manual del usuario.

Fase 8: Conversión de datos

Esta actividad requiere como entrada la base de datos actual del usuario y la especificación producida en el diseño. Aunque es sabido que en algunos proyectos la conversión de bases de datos involucra más trabajo que el desarrollo mismo del programa, en algunos otros ni siquiera existe una base de datos a convertir, tal es el caso de nuestro sistema en particular.

Fase 9: Instalación

Las entradas de esta actividad son: el manual del usuario, la base de datos y el sistema aceptado. En algunos casos la instalación pudiera significar simplemente un cambio al nuevo sistema de la noche a la mañana; en otros casos, la instalación pudiera ser un proceso gradual, en el que cada usuario va recibiendo manuales y entrenamiento, y comienza a usar el nuevo sistema.

1.3.2 Herramientas de modelado y planeación

En esta sección se discutirán de manera breve las herramientas que fueron utilizadas durante las etapas de análisis, diseño y desarrollo del sistema. En las etapas de análisis y diseño las herramientas descritas fueron seleccionadas del conjunto sugerido por la metodología de análisis y diseño estructurado, es decir, solo se presentan las que se adecuaron a nuestro problema en particular.

También durante el análisis se incluyen herramientas de planeación, las cuales fueron necesarias para establecer la trayectoria que el sistema debería seguir.

Durante la etapa de desarrollo o programación del sistema se empleó la metodología orientada a objetos, la cual resultó ser conveniente debido a la naturaleza del lenguaje de programación empleado y para manejar la información de los bancos.

1.3.2.1 Red de tareas

Una red de tareas muestra el orden en que las actividades de un proyecto deben ser realizadas. Puede mostrar también la gente, los productos, las herramientas requeridas, y como se crearán y combinarán productos intermedios para formar productos finales.

Permite verificar que cada tarea reciba sus entradas de tareas previas y mande sus salidas a tareas subsiguientes. La figura 1.3 muestra los componentes de una tarea. En caso de que una tarea no reciba sus entradas requeridas, estas deben ser creadas, pero si las salidas de la tarea no van dirigidas a ninguna otra, la tarea debe ser eliminada.

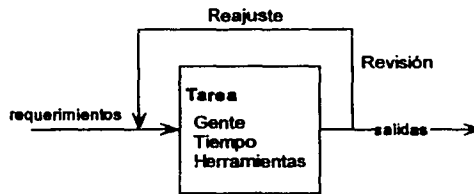


Figura 1.2 Componentes de una tarea.

Una red de tareas debe mostrar los puntos clave o alcances del proyecto, éstos pueden ser productos intermedios que serán revisados por el cliente, manuales, piezas de software, sistemas de ayuda. Estos forman un esqueleto inicial de la red. Después se agregan las tareas necesarias para proveer de entradas o para ser puntos de entrega y se unen a través de líneas para mostrar el orden de realización. Al final de cada tarea principal se pone una piedra miliar (diamante), indicando que todas las actividades previas deberán ser terminadas antes de proceder con cualquier tarea que se encuentre después de la piedra miliar. La figura 1.4 muestra un ejemplo de una red de tareas.

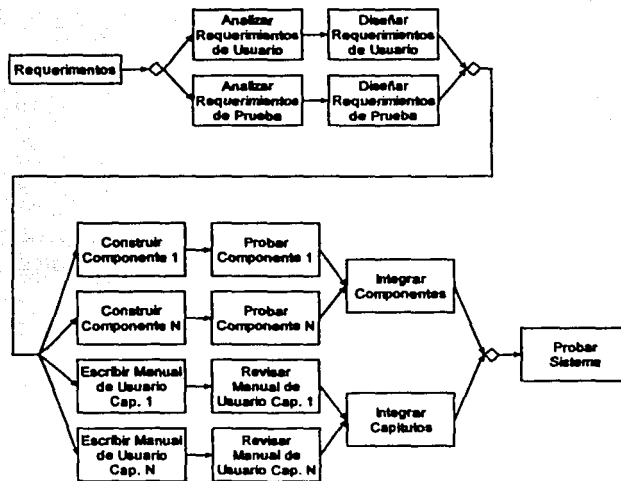


Figura 1.3 Red de tareas.

1.3.2.2 Diagrama de Gantt

El diagrama de Gantt muestra los tiempos asignados para la realización de las tareas en el proyecto. Los nombres de las tareas se escriben en una lista y los tiempos requeridos se establecen a continuación de cada tarea especificando los meses en los que se llevarán a cabo. La figura 1.5 muestra un ejemplo del diagrama de Gantt. Las líneas sólidas representan el plan original y las líneas punteadas el tiempo actual requerido. En el diagrama, la línea vertical representa el día actual, de tal manera que la tarea "Probar la rutina X" debería haber terminado, sin embargo, por retrasos en otras tareas, aún no se ha empezado. Las líneas con puntos separados, extienden el tiempo de realización de la tarea a una fecha de terminación proyectada.

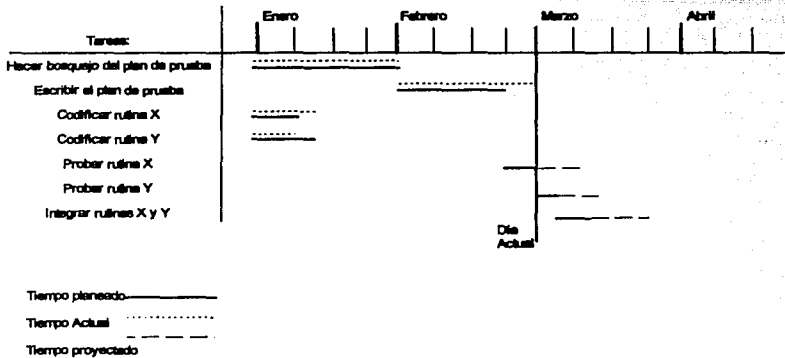


Figura 1.4 Diagrama de Gantt.

1.3.2.3 Diagrama de flujo de datos

El diagrama de flujo de datos (DFD) es una herramienta que permite visualizar a un sistema como una red de procesos funcionales conectados entre sí por "conductos" y "tanques de almacenamiento" de datos. Los DFDs son comúnmente usados en el desarrollo de sistemas operacionales, en donde las funciones del sistema son de gran importancia y son más complejas que los datos que maneja. La figura 1.6 muestra un ejemplo de un DFD.

TESIS CON
FALLA DE ORIGEN

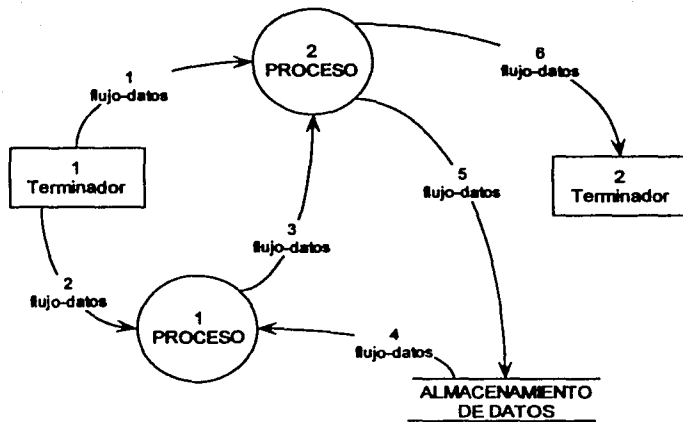


Figura 1.5 Diagrama de Flujo de Datos.

Los componentes de un diagrama de flujo de datos son:

- **Proceso.** Los procesos o burbujas muestran las partes del sistema que transforman entradas en salidas, se representan gráficamente a través del círculo y se nombran con una sola palabra, frase u oración sencilla, la cual describe el tipo de transformación que realizará sobre los datos.
- **Flujo.** El flujo de datos se representa por medio de una flecha que entra o sale de un proceso, se usa para describir el movimiento de bloques o paquetes de información de una parte del sistema a otra. Su nombre representa el significado del paquete que se mueve a lo largo del flujo.
- **Almacén.** El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas. El nombre que se utiliza para identificar al almacén, es el plural del que se utiliza para los paquetes que entran y salen por medio de flujo de datos.
- **Terminador.** Los terminadores se representan con el rectángulo. Estos representan entidades externas con las que se comunica el sistema. Algunos ejemplos de terminadores son personas, organizaciones externas, grupos o departamentos que estén dentro de la misma compañía u organización, pero fuera del control del sistema que se está modelando. En algunos casos un terminador puede ser otro sistema.

1.3.2.4 Diccionario de datos

El diccionario de datos es un repositorio de las definiciones de todos los flujos de datos y almacenamientos encontrados en un diagrama de flujo de datos, en una especificación de procesos, o en el mismo diccionario de datos, define datos compuestos en términos de sus componentes, y datos elementales en términos del significado de cada valor que pueden asumir. La tabla 1.1 muestra el significado de la notación usada en un diccionario de datos.

Símbolo	Significado
=	Está compuesto
+	Y
()	Opcional
{ }	Iteración
[]	Selección
**	Comentario
@	Identificador para almacenamientos
	Separador de opciones en una selección

Tabla 1.1 Notación empleada en un diccionario de datos.

1.3.2.5 Miniespecificaciones

Las miniespecificaciones o especificación de procesos es la declaración de las políticas que gobiernan las transformaciones de los flujos de datos de entrada en flujos de datos de salida en las burbujas del nivel inferior en un DFD (conocidas como primitivas funcionales). Algunos puntos importantes de las miniespecificaciones son:

- Debe existir una miniespecificación por cada primitiva funcional en un DFD.

- La miniespecificación debe establecer la manera en que los datos que entran a la primitiva son transformados en los datos que salen de ella.
- Debe establecer la política que gobierna la transformación de los datos, mas no el método de implementación (decir el qué y no el cómo).
- Debe buscar el control de la redundancia y no redefinir lo que ya ha sido definido en un DFD o en el diccionario de datos.
- El conjunto de constructores usados para crear una miniespecificación debe ser pequeño y simple.

Existen diversas herramientas usadas para la especificación de procesos tales como: inglés estructurado o pseudocódigo, árboles de decisión y tablas de decisión.

1.3.2.6 Diagrama de estructuras

El diagrama de estructuras ilustra la partición de un sistema en módulos (cajas negras) mostrando su jerarquía, organización y comunicación. Algunas de las ventajas del diagrama de estructuras son:

- Gráfico. Las herramientas gráficas son más fáciles de entender que las verbales.
- Divisible. Muestra en el nivel alto las funciones más importantes del sistema sin especificar detalles, estos son mostrados en los niveles inferiores.
- Riguroso pero flexible. Describe de manera precisa las funciones del sistema mostradas como cajas negras, pero como la descripción se mantiene por arriba del nivel del código, estas están sujetas a posibles cambios.
- Ofrece documentación para el sistema. Sirve como una herramienta de documentación útil durante la etapa de diseño.
- Ayuda en el mantenimiento y modificación. Ofrece un mapa del sistema que puede servir para encontrar errores.

La figura 1.7 muestra un ejemplo de un diagrama de estructuras.

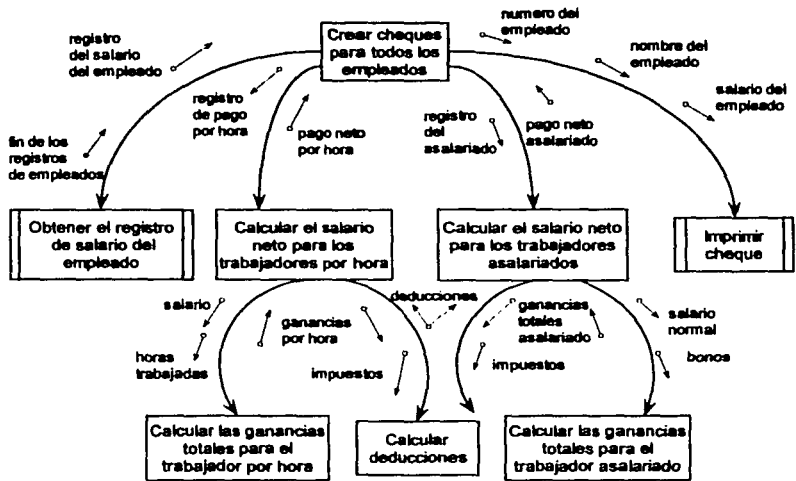


Figura 1.6 Diagrama de estructuras.

Los componentes del diagrama de estructuras son:

Módulo. Es un conjunto de directivas de programa con los siguientes atributos: entradas y salidas, función, mecanismos y datos internos. Las entradas y salidas son los datos que requiere y provee respectivamente. La función del módulo es lo que hace con las entradas para convertirlas en salidas. Los mecanismos son los procedimientos o lógica utilizados para realizar su función. Y los datos internos son los datos privados utilizados solo dentro del módulo. Gráficamente el módulo se representa mediante un rectángulo y su nombre describe la función que realiza. Un módulo predefinido se muestra con líneas paralelas a las líneas verticales del rectángulo.

Conexión entre módulos. Los módulos se conectan a través de flechas las cuales describen la llamada a una subrutina u otro módulo, la flecha indica cual módulo llama a cuál.

Comunicación entre módulos. Los módulos se comunican a través de los datos que intercambian. La flecha con el círculo vacío es llamada *conjunto de datos* y representa los datos a ser procesados. La flecha con círculo sólido es llamada *bandera*, esta representa información de los datos como su estado.

1.3.2.7 Interfaz gráfica del usuario (GUI)

Según Allan Cooper [Cooper 1995], las cualidades que hacen que una interfaz gráfica sea buena están centradas en el usuario y no en la tecnología. Dos de las cualidades más importantes que son centradas en el usuario son: visualidad del software y el vocabulario del programa.

La razón por la que los seres humanos procesamos más rápido y más eficientemente la información visual que la textual, reside en la habilidad de nuestro subconsciente de agrupar cosas en patrones basándose en indicaciones visuales. El diseño de interfaces gráficas trata de entender y aplicar este modelo de procesamiento de la información visual, en una filosofía que presente los componentes de los programas en la pantalla, como patrones visuales reconocibles, el texto agregado sirve solamente como un suplemento descriptivo.

En cuanto al vocabulario del programa, Alan Cooper sugiere restringir el rango del vocabulario usado para comunicarse con el usuario, en un conjunto de elementos atómicos. Entre mas elementos atómicos existan en el vocabulario, mayor el tiempo consumido y la dificultad para aprender a usarlo.

Todos los sistemas de comunicación fáciles de aprender obedecen el patrón llamado vocabulario canónico, el cual es representado a través de una pirámide invertida como se muestra en la figura 1.8.

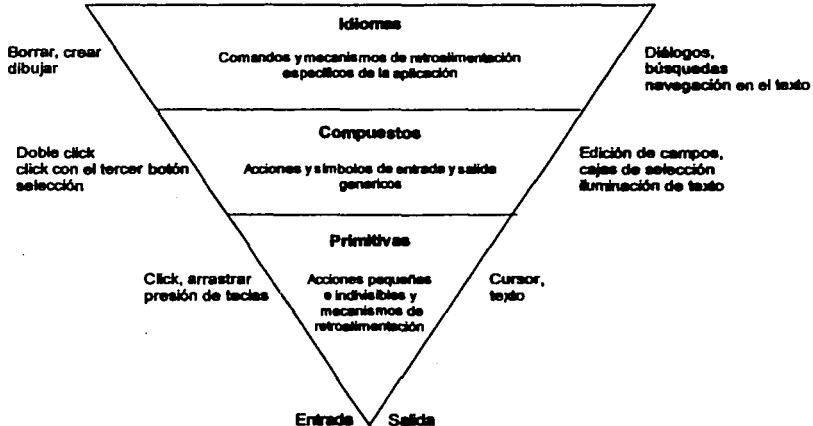


Figura 1.7 Vocabulario Canónico.

El segmento inferior de la pirámide contiene las *primitivas* o elementos atómicos a partir de los cuales se crea el lenguaje. En la parte intermedia están los *compuestos*. Estos son elementos más complejos creados al combinar dos o más primitivas, en el GUI estos son acciones tales como: doble-click con el mouse, click y arrastre, y objetos manipulables como botones y cajas de selección.

La parte superior de la pirámide contiene los *idiomas*. Los idiomas combinan compuestos con conocimientos del problema en consideración, conocidos como *conocimiento del dominio*. El conocimiento del dominio es información relacionada con el área de aplicación del usuario y no con la solución computarizada del problema. El conjunto de idiomas abre el vocabulario a información acerca del problema en particular que se trata de resolver, dentro del GUI éstos pueden ser: botones de aceptación, barras de captura, cajas de listas e íconos de archivos.

Según Cooper para crear interfaces visuales efectivas, se debe crear interacción con el usuario a través de un vocabulario formado canónicamente, que sea expresado visualmente y enfocado en los objetivos e intereses del usuario.

1.3.2.8 Enfoque orientado a objetos

El enfoque orientado a objetos aparece en 1968 como resultado de la necesidad de sistemas más flexibles, sobre todo para desarrollar interfaces gráficas más amigables, con un potencial y velocidad de ejecución más avanzado.

La programación orientada a objetos (POO) intenta modelar los objetos del mundo real con herramientas de programación. Se basa en la capacidad de abstraer las cosas que vemos, a través de sus propiedades tales como: tamaño, color, forma, peso, etc. y los diversos comportamientos que presentan: un bebé llora, duerme, juega y sonríe, una pelota rebota, rueda, se infla y poncha, una toalla absorbe agua, etc.

A las propiedades las conoce como *atributos* y a los diversos comportamientos los identifica como *métodos*.

La POO toma la ventaja de que algunas cosas comparten cierta similitud y las agrupa en conjuntos llamados "clases", de tal manera que una clase define el comportamiento y atributos de un grupo de objetos con características en común.

Por ejemplo, la figura 1.9 muestra la clase *auto* describiendo que todos los miembros que pertenezcan a esta clase presentan color, velocidad y carburador, y además pueden arrancar, ir, parar y girar.

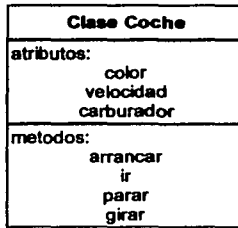


Figura 1.8 Ejemplo de una clase.

Un *objeto* se reconoce como una instancia de una clase, y se diferencia de otros miembros de la misma clase a través de sus atributos como lo muestra la figura 1.10.

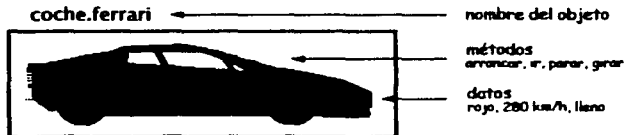


Figura 1.9 Instancia de una clase.

Una clase puede ser derivada de las características predefinidas de otra clase, a la clase derivada se le conoce como *subclase* y a la clase "padre" se le conoce como *superclase*.

Se sabe que las principales propiedades de la POO son: el encapsulamiento, la herencia y el polimorfismo.

El encapsulamiento. Los objetos ocultan íntimamente sus atributos y métodos, y aunque saben como comunicarse con otros objetos a través de interfaces bien definidas, no conocen de su implementación. Efectivamente es posible manejar un auto sin la necesidad de conocer los detalles del funcionamiento del motor, las transmisiones y demás.

La herencia. Es un mecanismo que permite la definición de una clase a partir de la definición de otra ya existente. Permite su refinamiento o especialización, de tal manera que se pueden definir, ocultar o añadir nuevos miembros. Por ejemplo, la clase de *coches convertibles* hereda todas las cualidades de la clase *coche*, y agrega la ventaja de que el techo se levanta y baja. La herencia es la característica que permite propiciar la reusabilidad.

El polimorfismo. Tal como la herencia, el polimorfismo permite la reusabilidad de atributos y métodos. Sin embargo, el polimorfismo introduce el concepto de *clases abstractas* las cuales permiten generalizar comportamientos o características. Las clases abstractas son tan generales que no pueden ser instanciadas. El propósito principal de una clase abstracta es proveer una superclase de la cual otras clases puedan heredar la implementación y/o interfaz. El polimorfismo es la característica que permite propiciar la extensibilidad.

1.3.3 Metodología de Análisis Estructurado

Como ya dijimos, Edward Yourdon [Yourdon, 1989] basa su metodología en el uso de diagramas para modelar los requerimientos del usuario y crear un sistema que le ayude a ser más productivo.

Yourdon sugiere la construcción de un Modelo Esencial en la etapa de análisis, dicho modelo se enfoca en lo que el sistema debe hacer para satisfacer los requerimientos del usuario, diciendo lo mínimo posible acerca de como se implanta.

Esto significa que cuando se hable con el usuario acerca de los requerimientos del sistema se debe evitar describir implantaciones específicas de procesos (burbujas en un diagrama de flujo de datos); es decir, no se deben mostrar las funciones del sistema que están siendo realizadas por humanos o por otros sistemas de cómputo existentes.

El modelo esencial consiste de dos componentes principales:

- 1 Modelo ambiental
- 2 Modelo funcional

1.3.3.1 El modelo ambiental

Según Yourdon la labor más difícil en la especificación de un sistema es determinar que va a ser parte del sistema y que no.

De tal manera que el primer modelo que se debe desarrollar es uno que sólo defina las interfaces entre el sistema y el resto del universo, es decir el ambiente. Por lo tanto, se necesita saber que información entra al sistema desde el ambiente exterior, y que información produce como salida al ambiente externo. A este modelo se le conoce como el *modelo ambiental*.

Un aspecto importante del modelo ambiental consiste en identificar los acontecimientos que ocurren en el ambiente a los cuales el sistema debe responder, no todos los acontecimientos se deben considerar, ya que el ambiente en su totalidad genera un número infinito de acontecimientos, solo nos preocupan aquellos que ocurren en el ambiente exterior y que requieren una respuesta del sistema.

Algunos de los factores a considerar para crear una perspectiva ambiental del proyecto son:

- El deseo del usuario por lograr cierta participación del producto en el mercado, o de incrementarla a más de su nivel actual. Esto se puede hacer ofreciendo un nuevo producto o una mayor funcionalidad del ya existente.
- La legislación establecida por el gobierno federal, estatal, o de la ciudad. Por ejemplo, tendría que hacerse un nuevo sistema para considerar los cambios en las leyes sobre impuestos.
- El deseo del usuario por lograr alguna ventaja estratégica para la línea de productos o áreas de negocios que opera.

El modelo ambiental consta de tres componentes:

- Declaración del propósito del sistema
- Diagrama de contexto
- Lista de eventos

La *declaración del propósito del sistema* consiste en la declaración textual breve y concisa de los objetivos que el sistema quiere cumplir, dirigida al nivel administrativo superior, a la administración de los usuarios, y a otros que no están directamente involucrados con el desarrollo del sistema.

El *diagrama de contexto* es un caso especial del DFD en donde una sola burbuja representa todo el sistema.

El diagrama de contexto enfatiza varias características importantes:

1. Las personas, organizaciones y sistemas con los que el sistema se comunica. Éstos se conocen como terminadores.
2. Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
3. Los datos que el sistema produce y que se envían al mundo exterior.
4. Los almacenes de datos que el sistema comparte con los terminadores. Estos almacenes de datos se crean fuera del sistema para su uso, o bien son creados en él y usados fuera.
5. La frontera entre el sistema y el resto del mundo.

La *lista de eventos* es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder.

Una vez definidos los componentes del modelo ambiental, se debe empezar la elaboración de un diccionario de datos.

1.3.3.2 El modelo funcional

El modelo funcional describe como el sistema debe operar en su ambiente. Se construye un diagrama de flujo de datos inicial a partir de la lista de eventos de la siguiente manera:

1. Se dibuja una burbuja o proceso por cada evento en la lista.
2. Se nombra la burbuja describiendo la respuesta que el sistema debe dar al evento asociado.
3. Se dibujan las entradas y salidas correspondientes de tal manera que la burbuja pueda dar la respuesta requerida, y se dibujan los almacenes de datos para la comunicación entre ellas.
4. Se compara el DFD resultante con el diagrama de contexto y la lista de eventos para asegurar que está completo y sea consistente.

El proceso de refinamiento del diagrama de flujo de datos incluye la realización de niveles hacia arriba y hacia abajo.

En los niveles hacia arriba se agrupan varias burbujas en una sola, manteniendo juntas aquellas que operan con datos comunes o que comparten un almacenamiento de datos, de tal manera que cada DFD al final contiene siete mas/menos dos burbujas.

En la refinación hacia abajo se divide cada burbuja en varias burbujas y almacenamientos de datos. Esto crea niveles inferiores y agrega mayor detalle. Se crean tantos niveles como sea necesario hasta que cada burbuja sea indivisible.

Al finalizar el DFD se completa el diccionario de datos agregando los nuevos datos y almacenamientos de datos.

Por cada burbuja en el último nivel del DFD se define una especificación de procesos o miniespecificación. Existen diversas herramientas para explicar lo que hace un proceso tales como: pseudocódigo, tablas y diagramas de flujo.

Se finaliza el modelo funcional creando un diagrama de transición de estados. Este define los estados importantes y las transiciones existentes en el sistema.

1.3.4 Metodología de Diseño Estructurado

El diseño estructurado utiliza partes del modelo esencial para crear un diseño. Al igual que el análisis estructurado, el diseño estructurado emplea una serie de modelos: modelo de implementación del usuario, modelo de implementación del sistema y modelo de implementación del programa.

1.3.4.1 El modelo de implementación del usuario

El modelo de implementación del usuario incluye las actividades entre el análisis de requerimientos y el diseño. Algunas de las actividades son:

1. Distribución del modelo esencial entre personas y máquinas. Esto significa determinar la frontera de automatización, es decir, definir que partes de las actividades del modelo esencial quiere el usuario que se automaticen y cuales quiere que funcionen manualmente.
2. Determinación de la interacción humano-máquina. Esta actividad incluye la selección de los dispositivos de entrada y salida, la elaboración de los formatos de todas las entradas que fluyen desde los terminadores hasta el sistema y de todas las salidas que fluyen desde el sistema hacia los terminadores, la definición de la secuencia y de los tiempos de las entradas y salidas en el sistema, y el diseño de la interfaz gráfica (GUI).
3. Consideración de las restricciones operativas que el usuario impone al sistema. Esto incluye el volumen de datos que el sistema debe manejar, los tiempos de respuesta esperados, las restricciones políticas y ambientales, y la confiabilidad de operación que el sistema debe otorgar. Estas restricciones limitan el espacio solución del diseño.

1.3.4.2 El modelo de implementación del sistema

El modelo de implementación del sistema inicia la etapa real del diseño. Se compone de los modelos de procesadores y de tareas.

El modelo de procesadores tiene como objetivo decidir cómo se distribuirá el modelo esencial en diferentes procesadores (CPUs) y definir la comunicación entre ellos. Existen diversas posibles configuraciones tales como el uso de un solo procesador. También se puede asignar un procesador para cada burbuja del modelo esencial, esta solución es conocida como la solución distribuida. Una tercera alternativa es el empleo de varias computadoras con diversas arquitecturas.

El modelo de tareas asigna procesos y almacenamientos de datos a tareas individuales dentro de cada procesador.

1.3.4.3 El modelo de implementación del programa

En el modelo de implementación del programa se llega al nivel de una tarea individual. Dentro de una tarea la computadora opera de una manera sincronizada: solo se puede llevar a cabo una actividad a la vez. El modelo más común de organización de las actividades en una unidad sincronizada es el diagrama de estructuras, que muestra la organización jerárquica de los módulos dentro de una tarea. Debe existir un diagrama de estructuras por cada tarea dentro de cada procesador.

Capítulo 2

ANÁLISIS

En este capítulo se muestran las actividades de análisis que se realizaron. Se llevaron a cabo varias sesiones de trabajo con los operadores del sistema actual y con los encargados de correr el servicio FixBan, de lo cual se obtuvo la información necesaria para generar una especificación estructurada, la cual será empleada como base para el diseño del sistema.

2.1 Estudio preliminar

La primera actividad de análisis que se realizó fue el estudio preliminar. Se realizaron cuatro juntas de trabajo, cuyos resultados se describen en la siguiente sección.

Con el estudio preliminar se lograron varios objetivos de los cuales se mencionan los más importantes:

- Al platicar con el usuario se pudieron definir algunas de las limitaciones del sistema actual, así como características a su favor.
- Cuando se vio al usuario operando el sistema se pudo confirmar las limitaciones previamente señaladas y vislumbrar posibles mejoras.
- Se comprendió el porqué de la existencia del sistema, y su interacción con otros sistemas a su alrededor.
- Se identificaron las personas que están involucradas o dependen del funcionamiento del sistema.
- Se obtuvo información suficiente para realizar una primera versión de los requerimientos del sistema.

2.1.1 Juntas de trabajo

JUNTA DE TRABAJO #1. La primera entrevista se dio en la Subgerencia de Cambios Nacionales del Banco de México, con los operadores del sistema actual. Se presenciaron las 3 sesiones para el cálculo del FIX, que se llevan a cabo en el día y se explicó la dinámica del proceso de las subastas. Los siguientes son los resultados que se obtuvieron:

- Se realizan tres sesiones o subastas en el lapso entre las 9:00 horas y las 12:00 horas.

- La primera subasta puede iniciar a partir de las 9 en punto. El operador se conecta al servicio del FixBan a través del cliente Banxico unos minutos antes. El sistema es capaz de informarle que bancos están conectados. Entonces la subasta se realiza como sigue:
 - El operador configura la subasta, especificando principalmente el monto a subastar (en dólares) y el número de participantes (al menos cuatro).
 - El operador abre la sesión, aparece una pantalla indicando a todos los participantes que la sesión ha iniciado.
 - El contador del tiempo de la subasta aparece indicando el tiempo restante.
 - Las posturas de compra y venta de los participantes empiezan a llegar. Solo los bancos elegidos pueden enviar posturas.
 - Se forma una gráfica de oferta y demanda con las posturas recientes de los participantes.
 - Al terminar el tiempo de la subasta (70 segundos), los participantes no pueden someter nuevas posturas.
 - Se notifica que el tiempo de arbitrajes ha empezado.
 - El contador de tiempo indica 10 segundos para arbitrajes.
 - Si las posturas formaron arbitrajes, el operador puede seleccionar uno o todos. Pero solo dentro de los 10 segundos. Los arbitrajes seleccionados pasarán a ser las asignaciones de la subasta.
 - Al finalizar el tiempo de arbitraje se notifica que la sesión a terminado. Sin embargo, si no se recibió al menos una postura por participante, se esperará a que se completen el mínimo de posturas para dar por terminada la sesión. Las posturas que entren fuera del tiempo permitido, no serán consideradas para arbitrajes.
 - Al terminar la sesión, se recibe la notificación del resultado del cálculo del FIX de la sesión.
 - El operador imprime los resultados y espera a la siguiente subasta.
 - Las tres subastas siguen la misma dinámica, al finalizar la última se obtiene el FIX del día y el TCVM (tasa de valor medio) del día.
 - El operador imprime los resultados del día y los reportes.
 - Se libera el resultado que será publicado al día siguiente en Internet.

- Los reportes impresos consisten de lo siguiente:
 - Al finalizar cada subasta se imprime las posturas finales, especificando la clave de los bancos que las propusieron. Se incluye el resultado del FIX de la sesión y la fecha.
 - Un segundo reporte incluye las posturas y la gráfica resultante.
 - Un tercer reporte contiene las asignaciones (en caso de ocurrencia), especificando el monto y los bancos con los que se realizó la operación.
 - Al finalizar las tres sesiones se imprime el FIX del día, el TCVM del día y la fecha.
 - Se imprime una carta dirigida a Comunicación Social, donde se especifica la fecha y el valor obtenido del FIX del día.
 - Se imprime un comunicado para la Subgerencia de Cambios Nacionales especificando fecha y el valor obtenido del FIX del día.

- Estas son las observaciones y conclusiones que se hicieron:
 - Se identificaron los siguientes estados que el sistema podría tomar: *sesión pasiva, sesión activa, arbitrajes y esperando clientes*. En sesión pasiva el cliente Banxico puede saber los nombres de los bancos que están conectados, los que se han desconectado y los que no se han conectado en el día. Se puede cambiar la configuración de las subastas y cambiar el password, enviar mensajes de indicaciones a otros bancos, imprimir los resultados de la última sesión, realizar la selección de los participantes para la siguiente subasta y empezar una subasta, en cierto modo es el estado de preparación para una subasta. La sesión activa, esencialmente ocurre mientras el tiempo de la subasta está corriendo. Los arbitrajes ocurren mientras el tiempo de arbitrajes está activo (10 segundos). El estado esperando clientes, solamente sucede cuando no se recibieron todas las posturas a tiempo.
 - Los reportes de los resultados de las sesiones solo pueden imprimirse justo al finalizar la sesión y antes de empezar la siguiente, ya que los resultados se desechan y no pueden ser recuperados.
 - Los reportes de Comunicación Social y el comunicado se realizan con la ayuda de una aplicación independiente.
 - Se publican los resultados del FIX del día en Internet a través de un programa externo.

- o Se determinó que el usuario principal es la Subgerencia de Cambios Nacionales y que aunque se producen reportes para otras entidades, éstas no son parte del sistema.

JUNTA DE TRABAJO #2. Esta fue una junta de trabajo realizada en la Subgerencia de Desarrollo de Infraestructura Informática, con la persona que desarrolló el sistema en función (FixBan 1.3), y los encargados de proveer el servicio. Los resultados obtenidos en ésta sesión son los siguientes:

- El nuevo sistema deberá ser diseñado para operar con el protocolo de comunicación del FEC, desarrollado en la misma subgerencia, en el cual los clientes no se comunican directamente con el servidor, si no que el FEC sirve como enlace y filtro entre aplicaciones. Para lo cual, se obtuvo el documento con las especificaciones del FEC.
- Deberá ser un sistema de 32 bits y operar en ambiente Windows.
- Se obtuvo una copia de la versión actual del sistema (1.3), para familiarizarse con la dinámica de las sesiones.
- Se aclaró que aunque algunos datos de la configuración de las subastas están especificados en el Diario Oficial de la Federación [Diario Oficial 1996], se espera que en el nuevo sistema puedan ser configurables.
- Se obtuvo el documento con las especificaciones del algoritmo para el cálculo del FIX y algunos ejemplos de casos, donde se muestran cotizaciones de compra y venta con su gráfica correspondiente.
- Se obtuvieron los formatos para la carta a Comunicación Social y para el comunicado para la Subgerencia de Cambios Nacionales.

JUNTA DE TRABAJO #3. Se realizaron las siguientes aclaraciones.

- El FIX de la sesión no podrá ser calculado hasta que se hayan obtenido todas las posturas (al menos una por cada participante), sin importar que el tiempo de participación ya haya terminado.
- El FIX del día no podrá ser calculado hasta que se hayan realizado todas las sesiones del día (actualmente son 3).
- Las instituciones financieras podrán participar solamente una vez en las sesiones del mismo día.
- Los arbitrajes se forman cuando la demanda es mayor que la oferta, es decir, cuando existen cotizaciones que a la compra son mayores que algunas a la venta.

Como producto de las juntas con los usuarios se realizó una lista de las deficiencias encontradas en el sistema actual y las posibles mejoras:

DEFICIENCIAS:

- El mantenimiento del sistema se vuelve muy difícil por ser una aplicación de 16 bits.
- El sistema no soporta el protocolo de comunicación del FEC. Es necesario volver a diseñarlo para hacerlo compatible.
- El sistema por sí mismo no puede producir los reportes requeridos para Comunicación Social y para la Subgerencia de Cambios Nacionales.
- Los reportes que produce el sistema no proveen de información detallada sobre las subastas.
- El sistema no es capaz de mantener los detalles y resultados de sesiones anteriores, ni de varias sesiones a la vez.
- El sistema no es capaz de ser reconfigurado para que se conecte a una dirección IP diferente.
- La función de cambio de contraseña no previene que se utilicen passwords fáciles de adivinar.
- Los datos de las subastas no pueden ser reconfigurados.
- El sistema es incapaz de publicar los resultados en Internet, utiliza el apoyo de una aplicación externa.

MEJORAS:

- Los reportes pueden incluir información mas detallada sobre cada sesión, tal como: hora en que se aceptó cada postura, nombre y clave de los participantes, todas las posturas sometidas por los participantes.
- La lista de participantes además de proveer su nombre podría también proveer su número telefónico, el cual es usualmente requerido por el operador.
- La lista de participantes podría ser mejorada al emplear colores para diferenciar a los bancos que están conectados de los desconectados, en vez de usar un caracter (*) y un caracter (C) como lo hace el sistema actual. Los colores son visualmente más fáciles de distinguir.
- Se puede proveer de históricos detallados de cada sesión, accesibles en todo momento.
- El sistema puede mantener una bitácora de las últimas direcciones IP donde se ha conectado.
- La función para el cambio de contraseña puede ser mejorada para que solo permita el uso de passwords fuertes o difíciles de adivinar.
- Se puede permitir cambiar la configuración de las subastas, para soportar futuras actualizaciones en el rango de los datos.
- Se puede proveer de un módulo que publique los resultados en Internet.

- Se puede proveer de una marquesina para que los anuncios capten la atención de los bancos participantes.
- La función de selección de arbitrajes puede ser mejorada organizándolos jerárquicamente, para asegurar que el operador siempre vea el mejor arbitraje primero.

JUNTA DE TRABAJO #4. Después se identificaron y se discutieron los posibles módulos en los que podría dividirse el sistema:

- **Entrada al sistema.** Este es el módulo de inicio de operación del sistema. Se requiere que la aplicación le pueda indicar al operador las últimas direcciones IP donde se conectó con el FEC exitosamente.
- **Configuración de las subastas.** Este módulo se encargará de informar cuál es la configuración actual y deberá permitir ser cambiable.
- **Manejo de participantes.** Este módulo manejará una lista con datos de las instituciones financieras participantes en las subastas. Esta lista es la que será empleada para elegir los candidatos a participar en las sesiones del día.
- **Ejecución de las subastas.** Este módulo se encargará de controlar la dinámica de las subastas.
- **Manejo de arbitrajes y asignaciones.** En caso de la ocurrencia de arbitrajes, este módulo se encargará de mostrarlos por orden de conveniencia para el Banco de México y facilitar la rápida realización de asignaciones.
- **Generación de reportes.** Este módulo se encargará de generar los reportes de cada sesión, así como las cartas requeridas por Comunicación Social y por la Subgerencia de Cambios Nacionales.
- **Publicación en Internet.** Este módulo se encargará de que se envíen los resultados a Internet.
- **Histórico de las subastas.** Este módulo proporcionará información histórica de cada una de las subastas realizadas en el día.
- **Cambio de password.** Este módulo se encargará de la configuración del password del operador.
- **Distribución de indicaciones.** Este módulo permitirá distribuir indicaciones a los bancos conectados al servidor.

2.1.2 Especificación de requerimientos

Los requerimientos son las funciones que se deben realizar para satisfacer las necesidades del usuario. Aunque se generan con los resultados obtenidos del estudio preliminar, estos van evolucionando conforme avanza el proyecto. Básicamente los requerimientos le dicen al diseñador que debe realizar el sistema y como le gustaría al cliente ver realizadas esas funciones.

A continuación se describen los requerimientos del sistema FixBan 2.0 cliente Banxico por módulos:

1. *Módulo de Comunicaciones.* Su objetivo es manejar la comunicación bidireccional con el FEC. Los requerimientos para este módulo son:
 - Mantener la conexión con el FEC.
 - Manejar los mensajes de desconexión del servidor y del FEC.
2. *Módulo de entrada.* Se encargará de obtener los datos de acceso para establecer la conexión con el FEC. Los requerimientos de este módulo son:
 - Mostrar la pantalla de inicio especificando nombre y versión del sistema.
 - Captura de datos de entrada:
 - Login = BANXICO, dato preestablecido y no cambiabile.
 - Dirección IP del FEC, dato preestablecido y cambiabile. Se requiere proveer una lista de las últimas direcciones en las que se ha conectado exitosamente.
 - Password, máximo 10 caracteres.
 - Botón para aceptar.
 - Botón para cancelar.
 - La autenticación del operador se hará a través del password con máximo tres intentos.
 - Intentar la conexión durante 10 segundos antes de enviar respuesta de conexión fallida.
 - Una vez conectado, mostrar el módulo principal.
 - En caso de falla de conexión identificar el problema.
 - En caso de falla mostrar los teléfonos de los ingenieros de mantenimiento del sistema.
3. *Módulo de configuración de subastas.* Su objetivo es permitir cambiar los datos con los que se ejecutarán las subastas y actualizarlos con el servidor. Los requerimientos para este módulo son:
 - Pantalla de datos con la configuración inicial:
 - Número de sesiones = 3
 - Número de participantes = 4
 - Monto = 5 millones
 - Divisa = dólar
 - Duración = 70 segundos
 - Decimales = 4

- Botón de guardar. Guardar la configuración significa actualizarla con el servidor.
 - Se debe validar los datos.
 - Se debe actualizar los datos con el servidor.
 - Botón de aceptar. Sale del módulo, permite cambiar los datos sin actualizarlos con el servidor.
 - Se debe validar los datos.
 - Se debe preguntar si se desea actualizarlos con el servidor, en caso contrario solo se debe reflejar el cambio en el cliente, sin embargo, al inicio de una subasta los últimos datos configurados deberán ser enviados al servidor.
 - Botón abrir sesión, será posible iniciar una sesión desde este módulo.
 - Se deberá verificar que los datos sean correctos y que el número de bancos seleccionados satisfaga el número de participantes, antes de iniciar la subasta.
 - Panel de información de valores y rangos. Mostrará los valores y rangos de cada dato de la subasta.
 - Número de sesiones, (valor) entero, (rango) $1 \leq X$.
 - Número de participantes, (valor) entero, (rango) $1 \leq X \leq \text{total de inscritos}$.
 - Monto, (valor) entero, millones, (rango) $0 \leq X$.
 - Divisa, (valor) dólar.
 - Duración, (valor) entero, segundos, (rango) $30 \leq X$.
 - Decimales, (valor) entero, (rango) $0 < X \leq 8$.
 - Se deberá indicar cuando alguno de los valores elegidos no está dentro del rango.
 - Se deberá informar cuando el servidor haya actualizado los datos, en caso de error se deberá mostrar los teléfonos de los ingenieros de mantenimiento del sistema.
4. *Módulo de manejo de participantes.* Proveerá la lista de las instituciones financieras que están inscritas en el cálculo del FLX, y los controles necesarios para seleccionar los participantes. Los requerimientos de este módulo son:
- Aparecer en la pantalla principal.
 - Mostrar una tabla con los nombres de las entidades financieras en orden alfabético.
 - Mostrar una tabla para alojar a los bancos seleccionados para participar.
 - Botones de selección y de-selección.
 - Permitir seleccionar / de-seleccionar uno o más bancos a la vez.
 - Actualizar con el servidor los bancos seleccionados / de-seleccionados.

- Mostrar contador del número de seleccionados.
 - Identificar con color el estado de los bancos, verde = conectado, rojo = desconectado, sin color = nunca conectado.
 - Se deberá mostrar un menú con: nombre completo, clave y teléfono de los bancos.
 - Se deberá diferenciar los bancos que ya participaron:
 - Ya cotizó ✓
 - No cotizó ✗
5. *Módulo de ejecución de subastas.* Su objetivo es mostrar la ejecución de las subastas. Este módulo deberá:
- Recibir las cotizaciones de los bancos.
 - Mostrar en una tabla: Nombre del banco, cotización de compra, cotización de venta.
 - Crear la gráfica de compra/venta con las últimas cotizaciones.
 - Mostrar en un reloj el tiempo restante de la subasta.
 - Avisar cuando el tiempo de participación haya terminado.
 - Si no se recibieron todas las posturas indicar que se está esperando clientes.
 - Informar cuando la subasta haya terminado.
6. *Módulo de arbitrajes y asignaciones.* Se encargará de mostrar los arbitrajes (en caso de que existan) y proporcionar los controles para realizar asignaciones. Este módulo deberá:
- Mostrar en una tabla todos los arbitrajes posibles por orden de importancia.
 - Permitir seleccionar uno o más arbitrajes.
 - Contar con un control para enviar los arbitrajes.
 - Permitir la realización de arbitrajes manuales y validarlos.
 - Enviar los arbitrajes al servidor.
 - Recibir la aceptación/cancelación de los arbitrajes del servidor.
 - Mostrar tiempo de arbitrajes.
7. *Módulo de reportes.* Deberá generar los siguientes reportes con los datos obtenidos del FIX del día:
- Carta para Comunicación Social.
 - Comunicado para la Subgerencia de Cambios Nacionales.

- Deberá permitir guardar e imprimir los reportes.
8. *Módulo de publicación en Internet.* Deberá solicitar al servidor la publicación del resultado del FIX del día en Internet.
 9. *Módulo de historias de las subastas.* Deberá mostrar el histórico de cualquier subasta realizada en el día. Se requiere que:
 - Permita elegir el número de la subasta cuya información se desea obtener.
 - Deberá solicitar los datos del histórico de la subasta al servidor.
 - Deberá recibir los datos y organizarlos en tablas y gráficas.
 - Deberá guardar e imprimir el histórico.
 - Los datos a mostrar son:
 - TCVM, FIX y fecha
 - Configuración de la subasta
 - Todas las posturas de los participantes (nombre, clave, postura de compra, postura de venta, hora)
 - Gráfica de compra y venta, con las últimas posturas obtenidas.
 - Arbitrajes.
 - Asignaciones.
 10. *Módulo de cambio de password.* Deberá permitir cambiar el password del operador. Sus requerimientos son los siguientes:
 - Mostrar una pantalla para cambio de password.
 - Solicitar password anterior.
 - Verificar password anterior.
 - Solicitar nuevo password.
 - Validar password: Palabra mayor a 6 letras formada con combinación de mayúsculas y minúsculas y/o números y/o caracteres especiales.
 - Confirmar nuevo password.
 - Botón aceptar.
 - Botón cancelar.
 - Enviar el nuevo password al FEC.
 11. *Módulo de distribución de indicaciones.* Este módulo deberá:

- Proveer de una pantalla de captura de mensajes para ser enviados a los bancos conectados.
- El texto podrá tener la forma de mensaje largo o cintilla móvil (marquesina).
- Deberá permitir seleccionar los bancos a los que va dirigido el mensaje.
- Deberá enviar el mensaje / cintilla al servidor para ser distribuido.
- Deberá indicar cuando el mensaje sea distribuido.
- En caso de error se deberá proveer el número telefónico de los ingenieros de mantenimiento del sistema.

12. *Módulo salir del sistema.* Permite finalizar la aplicación. Este módulo deberá:

- Indicar al FEC de su desconexión.
- Terminar la conexión.
- Terminar la aplicación.

13. *Módulo principal.* Tiene como objetivo permitir cargar el resto de los módulos:

- Módulo de manejo de participantes.
- Módulo de históricos de las subastas.
- Módulo de reportes.
- Módulo de publicación en Internet.
- Módulo para cambiar la configuración:
- Módulo para iniciar una sesión:
 - Indicar cuando se inicie la sesión y cargar el módulo de subastas.
 - Bloquear todos los controles durante una subasta excepto los del módulo de subastas.
 - Recibir el resultado del cálculo del FLX de la sesión y mostrarlo.
 - Recibir los resultados del FIX del día y TCVM del día y mostrados.
- Módulo para el cambio de password
 - Indicar si el password fue aceptado por el servidor o no.
- Módulo salir del sistema
- Mostrar en un panel de información: Estado de la sesión, número de la sesión, nombre del cliente, dirección IP del FEC, hora y fecha.

Adicionalmente se especificaron los requerimientos técnicos que se debían cumplir:

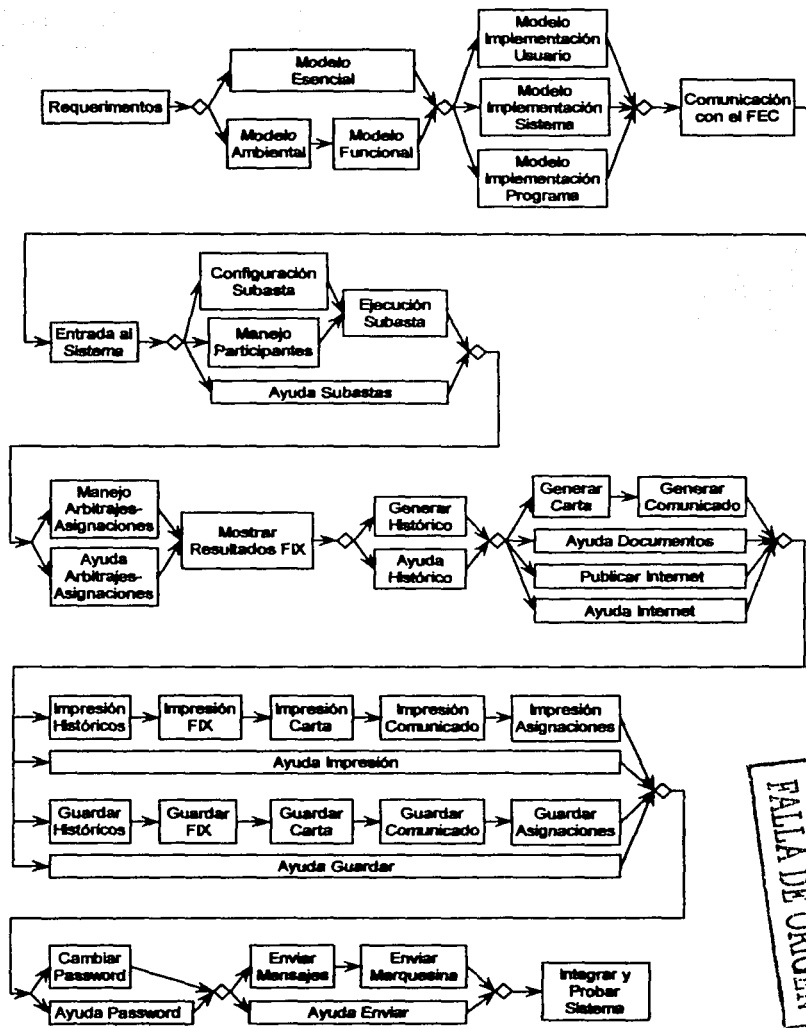
- Emplear el protocolo de comunicación del FEC.
- Usar el lenguaje de programación Visual Basic 5.0.

2.1.3 Planeación

Una vez especificados los requerimientos, se creó un plan para administrar los recursos con los que se contaba. Se creó una red de tareas para visualizar mejor el grado de importancia entre una y otra, y poder decidir la trayectoria a seguir. Después se creó un diagrama de Gantt para asignar tiempo para su realización.

Red de Tareas. La figura 2.1 muestra la red final de tareas, en ésta se incluye la especificación de requerimientos, análisis y diseño, los módulos del sistema y la documentación o manual del usuario. La etapa final que se realizó fue la integración de los módulos y las pruebas del sistema.

Los aspectos que se consideraron al realizar la red de tareas fueron el grado de dificultad de cada una y la dependencia existente entre unas y otras, de tal manera que se realizaron primero aquellas cuyas salidas eran las entradas de otras.



TESIS CON
 FALLA DE ORIGEN

Figura 2.1 Red de tareas del sistema FixBan cliente Banxico.

Diagrama de Gantt. La figura 2.2 muestra el diagrama de Gantt resultante. Se destinaron cuatro meses para la elaboración del proyecto. Las piedras miliare delimitaron las etapas de avance, las actividades previas a cada piedra miliar tenían que ser completadas antes de emprender una nueva fase.

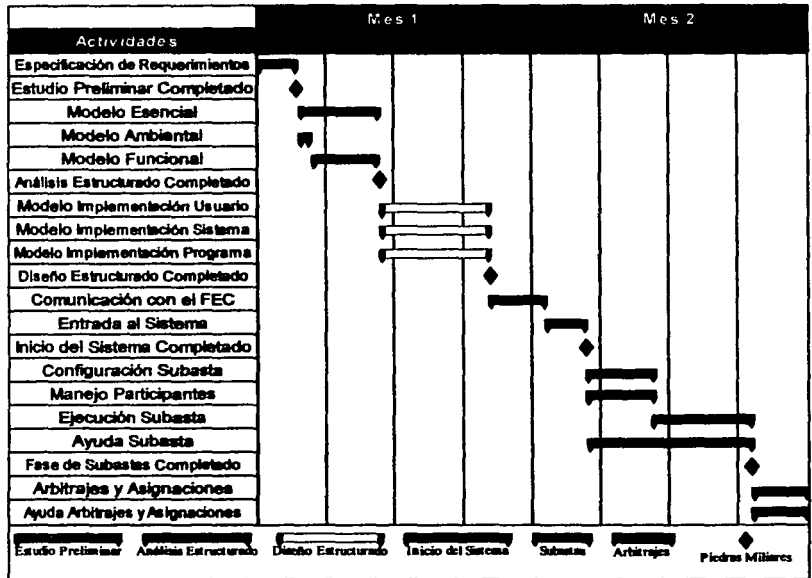


Figura 2.2 Diagrama de Gantt del sistema FixBan cliente Banxico.

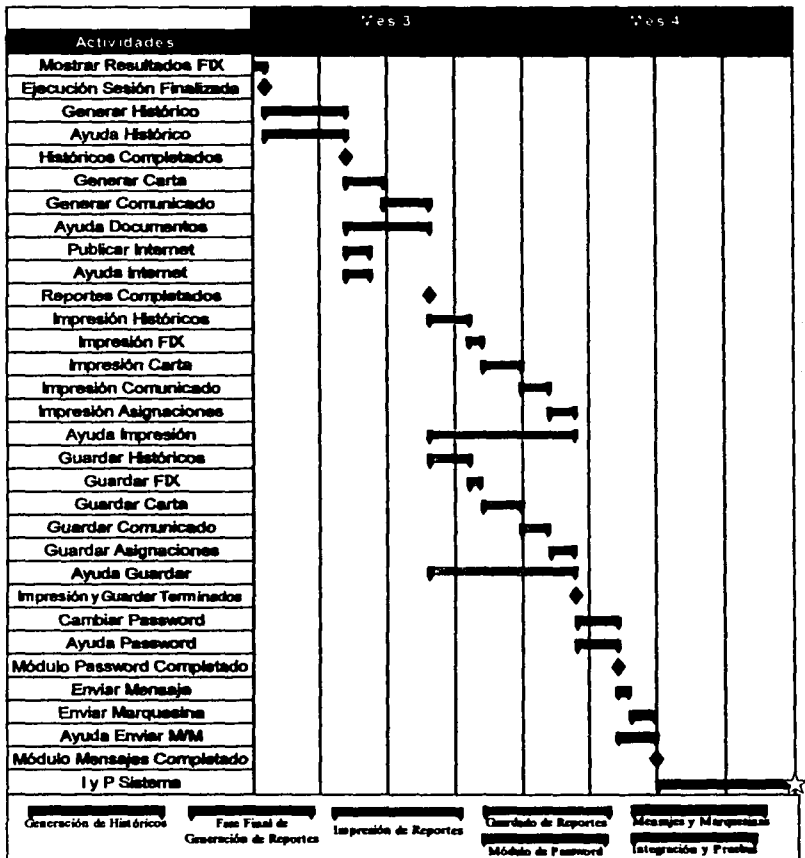


Figura 2.2 Diagrama de Gantt (continuación).

2.2 La especificación estructurada

Esta etapa analiza los requerimientos a través de diagramas y modelos del sistema, el resultado se conoce como la Especificación Estructurada.

2.2.1 El modelo esencial del sistema FixBan cliente Banxico

El modelo esencial es el principal producto de la especificación estructurada. Define dos modelos, el modelo ambiental y el modelo funcional. El primero identifica las fronteras del sistema y el segundo sus funciones.

2.2.2 El modelo ambiental del sistema FixBan cliente Banxico

El primer paso en la construcción del modelo ambiental es identificar las fronteras entre el medio ambiente y el proyecto objetivo. La figura 2.3 muestra las fronteras entre el cliente Banxico, el resto del sistema FixBan 2.0 y el medio ambiente.

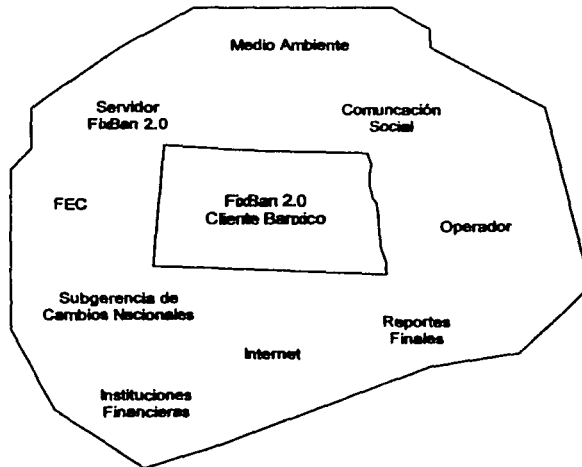


Figura 2.3 Fronteras del sistema.

El siguiente paso consiste en describir el propósito del sistema, crear el diagrama de contexto y una lista de eventos.

Propósito: "El propósito del sistema FibBan cliente Banxico es ser el centro desde donde se controlen las operaciones para el cálculo de la equivalencia del peso mexicano contra el dólar norteamericano, conocido como FLX."

Diagrama de contexto. La figura 2.4 muestra el diagrama de contexto del sistema. Se identificaron dos terminadores, el operador y el FEC de comunicaciones y un almacenamiento.

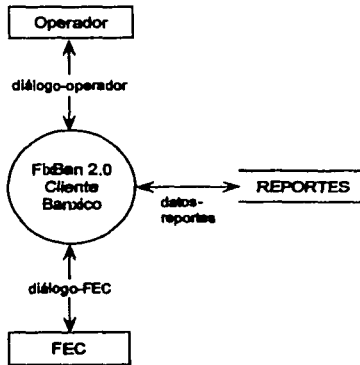


Figura 2.4 Diagrama de contexto.

El flujo de datos *diálogo-operador* consiste de las peticiones de inicio de subasta, configuración de las sesiones, selección de participantes y arbitrajes, impresión de reportes, publicación en Internet, generación de históricos, cambio de passwords y demás acciones que el operador requiera que se lleven a cabo.

El flujo de datos *diálogo-FEC* representa los mensajes entre el sistema Banxico y el FEC de comunicaciones, donde se le pide dirigir las peticiones del operador para ser ejecutadas por el servidor. Así mismo, los mensajes de respuesta provenientes del servidor y que pasan a través del FEC son parte del diálogo.

Lista de eventos. La lista de eventos son las acciones que el sistema va a realizar con los datos. Se identifica un evento por cada respuesta que el sistema deba dar. Tabla 2.1.

1. El operador solicita acceso.
2. El operador solicita cambio de password.
3. El operador configura una subasta.
4. El operador selecciona / de-selecciona bancos para la subasta.
5. El FEC indica que un banco se ha conectado/desconectado.
6. El operador solicita enviar una indicación en forma de "mensaje" o "marquesina" a una selección de bancos.
7. El operador solicita iniciar una subasta.
8. El FEC envía las posturas de los bancos participantes en la subasta.
9. El FEC envía lista de arbitrajes.
10. El operador selecciona arbitrajes para asignación.
11. El FEC informa el tiempo restante de la subasta. El FEC informa el tiempo restante de los arbitrajes.
12. El FEC indica que el tiempo de aceptación de posturas ha terminado.
13. El FEC envía mensaje de inicio de arbitrajes.
14. El FEC informa que el tiempo de los arbitrajes ha terminado.
15. El FEC informa que se están esperando clientes.
16. El FEC indica que la sesión ha finalizado.
17. El FEC envía los resultados del cálculo del FIX de la subasta.
18. El FEC envía los resultados del cálculo del FIX del día.
19. El operador solicita el histórico de alguna subasta.
20. El operador solicita la publicación del resultado del FIX del día en Internet.
21. El operador solicita la generación de cartas.
22. El operador solicita la historia de las asignaciones de las subastas.
23. El operador solicita la impresión de algún reporte.
24. El operador solicita el guardado de algún reporte.
25. El FEC informa que el FEC "ha muerto".
26. El FEC pregunta si el cliente "esta-vivo".

Tabla 2.1 Lista de eventos.

Para complementar la lista de eventos se empezó la redacción del diccionario de datos (apéndice A), donde se definieron los terminadores y componentes del diagrama de contexto.

2.2.3 El modelo funcional

El modelo funcional representa el comportamiento interno que debe tener el sistema para interactuar satisfactoriamente con el medio ambiente.

El primer paso fue construir un diagrama de flujo de datos, siguiendo la técnica de partición de eventos, dibujando un proceso o burbuja por cada respuesta del sistema que se identificó en la lista de eventos. La figura 2.5 muestra los DFD's resultantes.

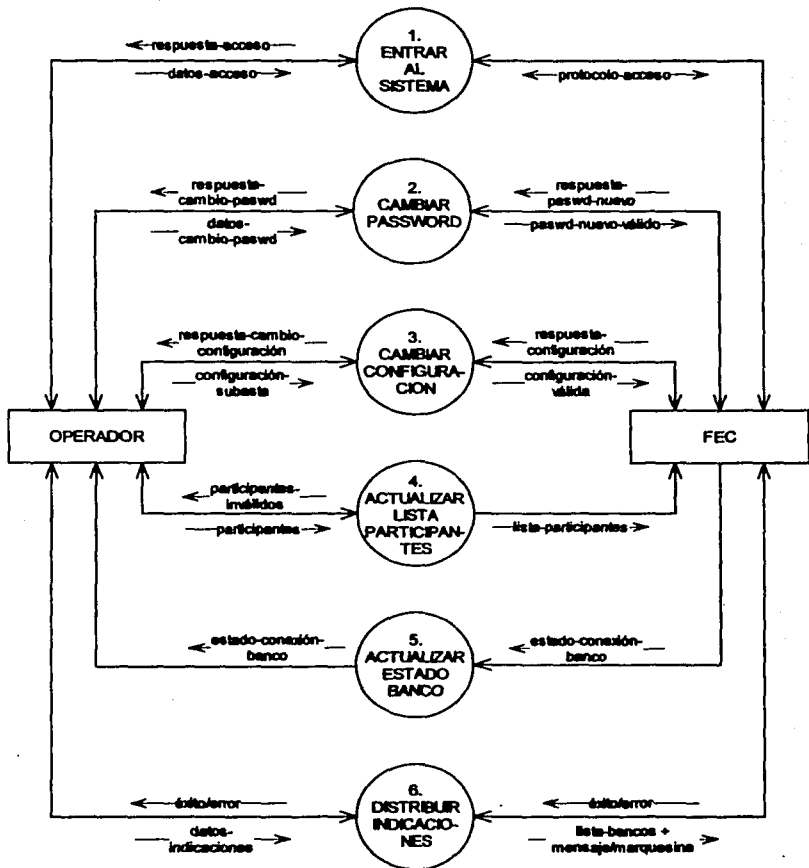


Figura 2.5 DFD de nivel 1.

TESIS CON
FALLA DE ORIGEN

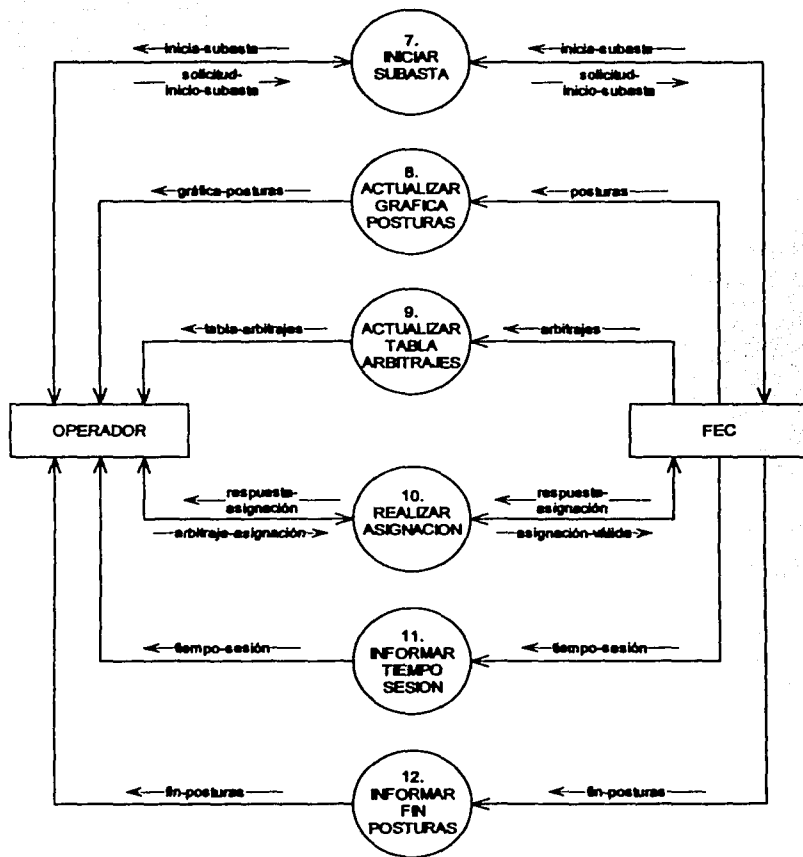


Figura 2.5 DFD de nivel 1 (continuación).

TESIS CON FALLA DE ORIGEN

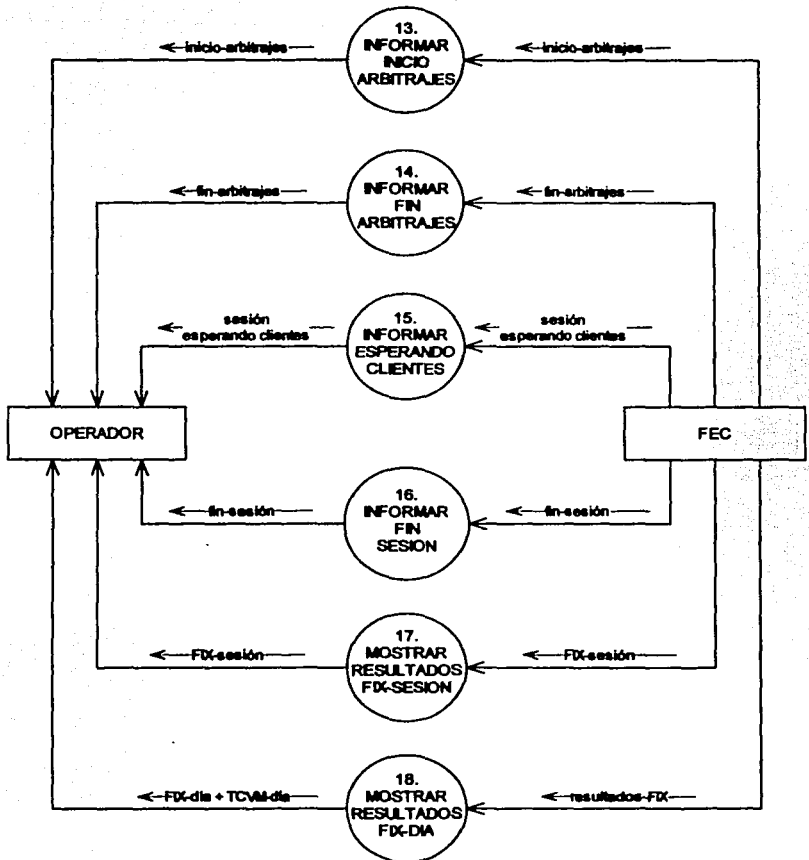


Figura 2.5 DFD de nivel 1 (continuación).

TESIS CON
FALLA DE ORIGEN

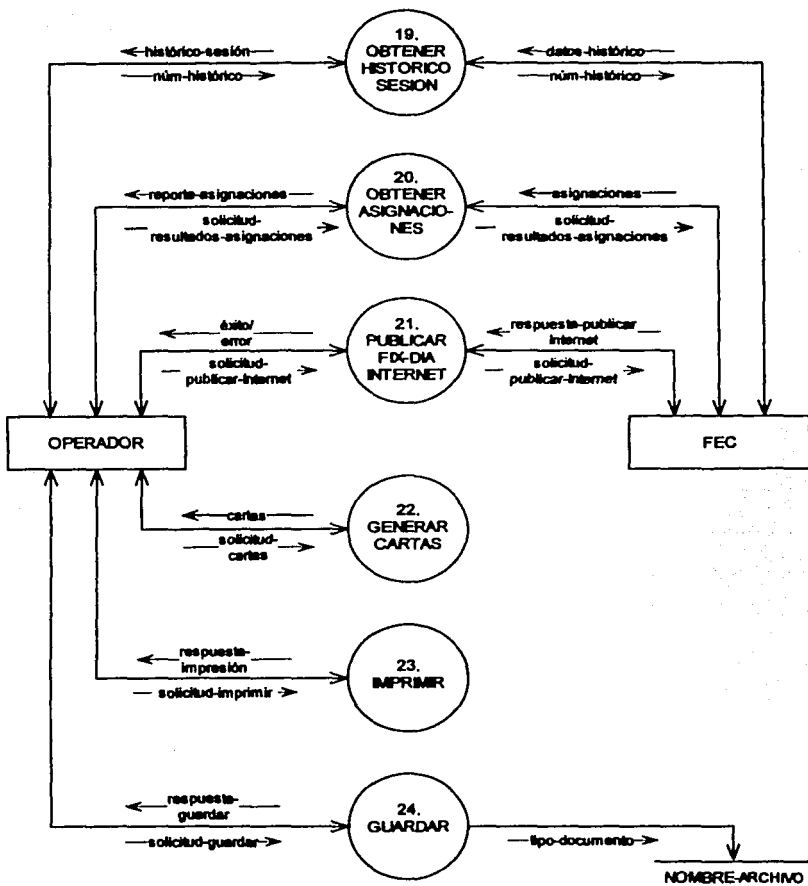


Figura 2.5 DFD de nivel 1 (continuación).

TESIS CON FALLA DE ORIGEN

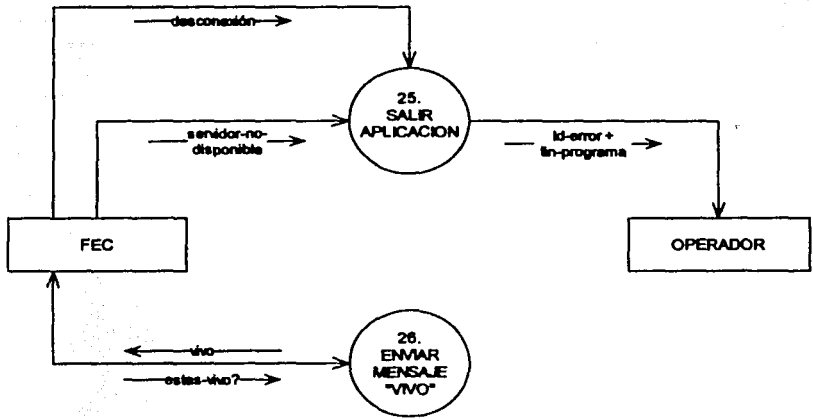


Figura 2.5 DFD de nivel 1 (continuación).

Para refinar el diagrama de flujo de datos inicial, primero se revisó la consistencia con el diagrama de contexto y la lista de eventos, después se realizaron diagramas de bajo nivel, los cuales se muestran en la figura 2.6. Cabe mencionar que aunque se sugiere realizar diagramas de alto nivel, en éste caso el diagrama de contexto representa el nivel justo arriba del diagrama de flujo de datos.

TESIS CON
FALLA DE ORIGEN

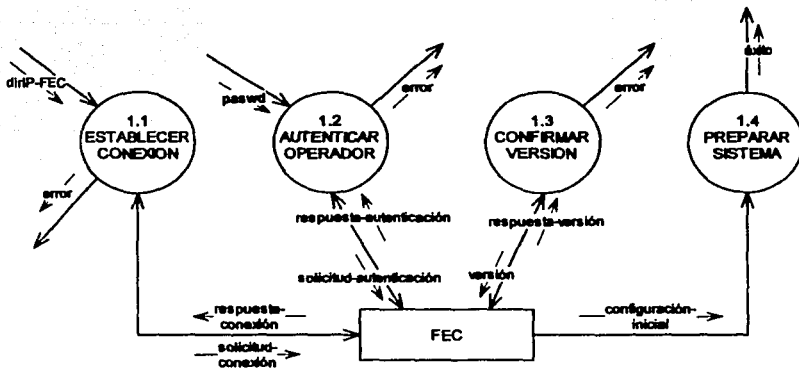


Figura 2.6 DFD de bajo nivel. Proceso 1 Entrar al Sistema.

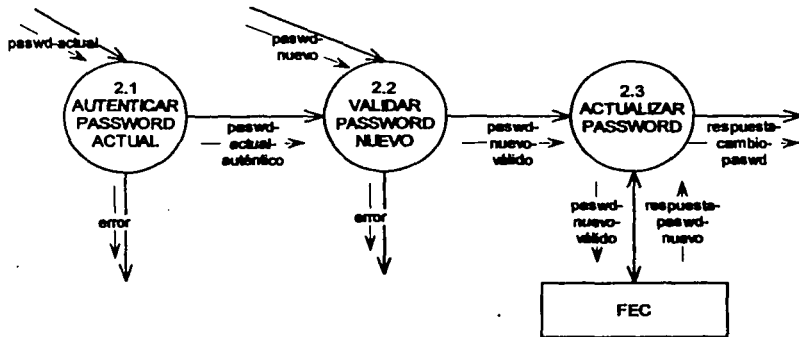


Figura 2.6 DFD de bajo nivel (continuación). Proceso 2 Cambiar Password.

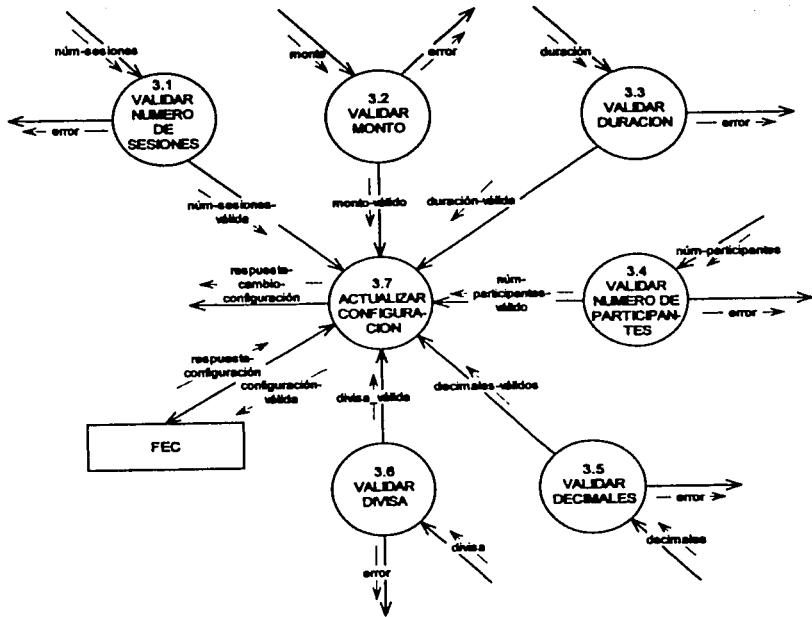


Figura 2.6 DFD de bajo nivel (continuación). Proceso 3 Cambiar Configuración.

TESIS CON
FALLA DE ORIGEN

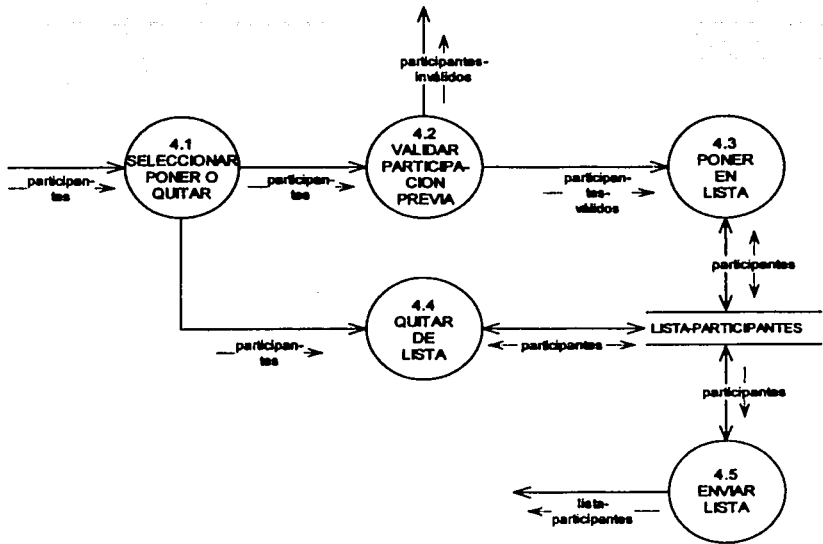


Figura 2.6 DFD de bajo nivel (continuación). Proceso 4 Actualizar Lista de Participantes.

TESIS CON
FALLA DE ORIGEN

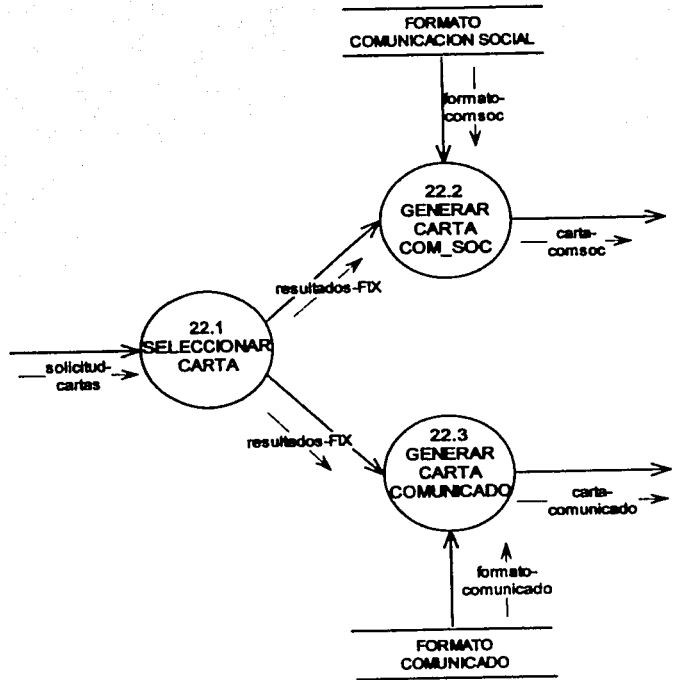


Figura 2.6 DFD de bajo nivel correspondientes al proceso (22) Generar Cartas.

TESIS CON FALLA DE ORIGEN

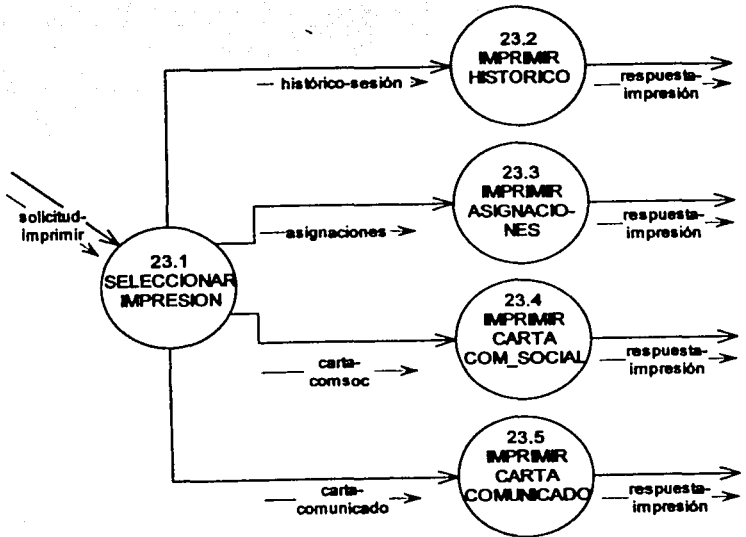


Figura 2.6 DFD de bajo nivel (continuación). Proceso 23 Imprimir.

TESIS CON
FALLA DE ORIGEN

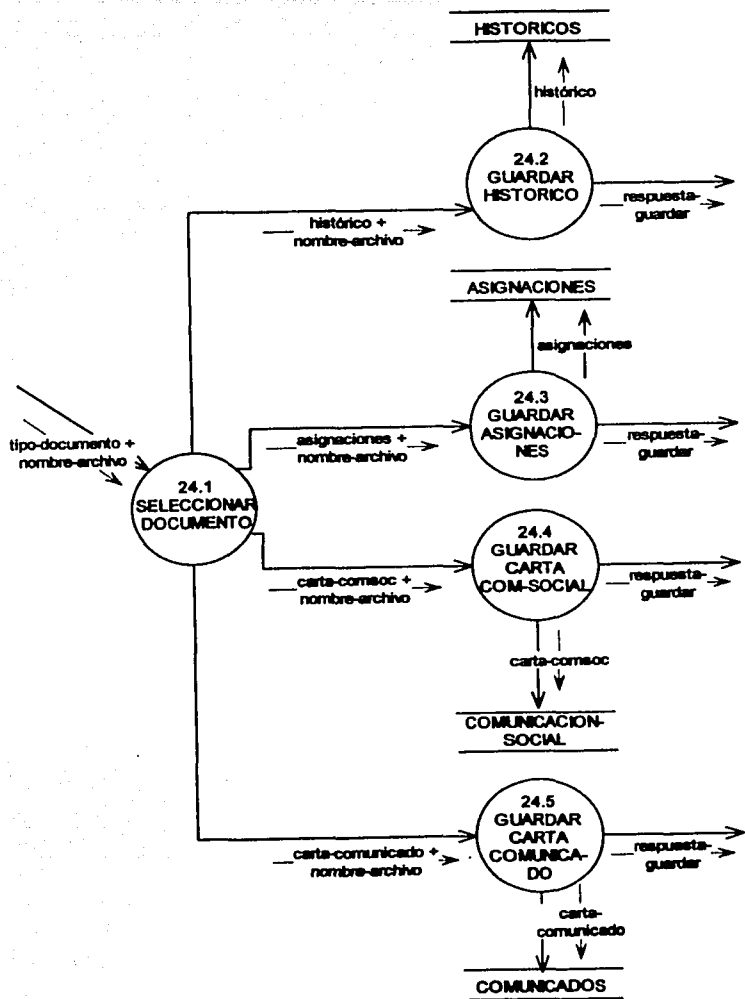


Figura 2.6 DFD de bajo nivel (continuación). Proceso 24 Guardar.

Una vez que se alcanzó el mayor detalle en los diagramas de bajo nivel, se crearon miniespecificaciones para cada proceso.

Proceso 1.1.1 ESTABLECER CONEXION

INICIO

RECIBE dirIP-FEC del operador

PARA CADA puerto HAS:

CONECTA CON dirIP-FEC EN puerto

SI NO RECIBE respuesta-conexión DEL FEC

ENVIA (al operador) respuesta-acceso = "Error
Conexión: No se pudo establecer comunicación
con el FEC"

SALIR DEL SISTEMA

//En caso de que respuesta-conexión sea éxito,
significa que el sistema podrá continuar su
ejecución

FIN SI

FIN PARA CADA

FIN

Proceso 1.2 AUTENTICAR OPERADOR

INICIO

RECIBE paswd del operador

PON login COMO "Banxico"

PON servidor COMO "FixBan"

núm-opportunidades = 0

REPITE

ENVIA PARA AUTENTICACION (al FEC) paswd, login,
servidor

OBTEN respuesta-autenticación DEL FEC

SI respuesta-autenticación = error

ENVIA (al operador) respuesta-acceso = "Error
Autenticación: El password es incorrecto,
trate de nuevo"

OBTEN otro-paswd DEL operador

PON paswd COMO otro-paswd

núm-opportunidades = num-opportunidades + 1

FINSI

HASTA (respuesta-autenticación = éxito) ó (núm-
oportunidades = 3)

SI respuesta-autenticación = error

ENVIA (al operador) respuesta-acceso = "Error:
Oportunidades agotadas"

SALIR DEL SISTEMA

FIN SI

//En caso de que respuesta-autenticación sea éxito,
significa que el sistema seguirá su ejecución
FIN

Proceso 1.3 CONFIRMAR VERSION

INICIO

PON versión COMO "2.0"

ENVIA PARA CONFIRMACION (al FEC) versión

RECIBE respuesta-versión DEL FEC

SI respuesta-versión = error

ENVIA (al operador) respuesta-versión = "Error:

Las versiones del cliente y el servidor no
coinciden"

SALIR DEL SISTEMA

FIN SI

//En caso de que respuesta-versión sea éxito,
significa que el sistema podrá
continuar su ejecución

FIN

Proceso 1.4. PREPARAR SISTEMA

INICIO

RECIBE lista-bancos DEL FEC

RECIBE configuración-subasta DEL FEC

RECIBE número-sesión DEL FEC

RECIBE estado-sesión DEL FEC

RECIBE estado-conexión-banco DEL FEC

RECIBE (si existen) posturas DEL FEC

RECIBE (si existen) arbitrajes DEL FEC

RECIBE (si existen) participantes DEL FEC

RECIBE (si existen) resultados-FIX DEL FEC

ENVIA (al operador) "éxito"

FIN

Proceso 2.1 AUTENTICAR PASSWORD ACTUAL

INICIO

OBTEN paswd-actual del operador

núm-opportunidades = 0

REPITE

ENVIA PARA AUTENTICACION (al FEC) paswd-actual

RECIBE respuesta-paswd-actual DEL FEC

SI respuesta-paswd-actual = error

ENVIA (al operador) respuesta-cambio-paswd =

"Error Autenticación: El password no coincide
con el password actual, trate de nuevo"

OBTEN (del operador) otro-paswd

PON paswd-actual COMO otro-paswd


```

        núm-oportunidades = núm-oportunidades + 1
    FIN SI
    HASTA (respuesta-paswd-actual = éxito) ó (núm-
oportunidades = 3)

    SI respuesta-paswd-actual = éxito
        ENVIA (al proceso 2.2) paswd-actual-auténtico
    SINO
        ENVIA (al operador) respuesta-cambio-paswd =
        "Error: Oportunidades agotadas"
    FINSI
FIN

Proceso 2.2 VALIDAR PASSWORD NUEVO
INICIO
    OBTEN paswd-nuevo DEL operador
    REPITE
        //Revisar que el password nuevo no sea el mismo
        que el actual.
        SI paswd-nuevo = paswd-actual
            PON paswd-inválido COMO VERDADERO
        FIN SI
        //Revisar que la longitud del password sea válida.
        SI LONGITUD paswd-nuevo <> longitud-permitida
            PON paswd-inválido COMO VERDADERO
        FIN SI
        SI paswd-nuevo <> paswd-fuerte
            PON paswd-inválido COMO VERDADERO
        FIN SI

        SI paswd-inválido = VERDADERO
            ENVIA (al operador) respuesta-cambio-paswd =
            "Error Validación: El password nuevo debe ser
            una combinación de mayúsculas, minúsculas, y
            números o caracteres-especiales, diferente
            al password anterior y de longitud de 5 a 8
            caracteres."
            OBTEN (del operador) otro-paswd
            PON paswd-nuevo COMO otro-paswd
        FINSI
    HASTA (paswd-inválido = FALSO)

    SI paswd-inválido = FALSO
        ENVIA (al proceso 2.3) paswd-nuevo-válido
    FIN SI
FIN

```

Proceso 2.3 ACTUALIZAR PASSWORD

INICIO

PON servidor COMO "FixBan"

ENVIA (al FEC) servidor, paswd-nuevo-válido

RECIBE respuesta-paswd-nuevo DEL FEC

SI respuesta-paswd-nuevo = éxito

ENVIA (al operador) respuesta-cambio-paswd =

Exito: "El password ha sido cambiado"

SINO

ENVIA (al operador) respuesta-cambio-paswd =

Error: "El password no pudo ser cambiado"

FIN SI

FIN

Proceso 3.1 VALIDAR NUMERO DE SESIONES

INICIO

OBTEN núm-sesiones

SI núm-sesiones \square 3 ENVIA (al proceso 3.7) núm-sesiones-válida

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Número de Sesiones Inválido: El número de sesiones debe ser mayor o igual a 3"

FIN SI

FIN

Proceso 3.2 VALIDAR MONTO

INICIO

OBTEN monto

SI monto \square 0 ENVIA (al proceso 3.7) monto-válido

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Monto Inválido: El monto debe ser mayor que cero"

FIN SI

FIN

Proceso 3.3 VALIDAR DURACION

INICIO

OBTEN duración

SI (duración \square 30) y (duración \square 70) ENVIA (al proceso 3.7) duración-válida

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Duración Inválida: La duración debe ser mayor o igual a 30 y menor o igual que 70"

FIN SI

FIN

Proceso 3.4 VALIDAR NUMERO PARTICIPANTES

INICIO

OBTEN núm-participantes

SI núm-participantes \square 4 ENVIA (al proceso 3.7) núm-participantes-válido

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Número de Participantes Inválido: El número de participantes debe ser mayor o igual a 4"

FIN SI

FIN

Proceso 3.5 VALIDAR DECIMALES

INICIO

OBTEN decimales

SI (decimales \square 4) y (decimales \square 8) ENVIA (al proceso 3.7) decimales-válidos

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Decimales Inválidos: El número de decimales debe ser mayor o igual a 4 y menor o igual a 8"

FIN SI

FIN

Proceso 3.6 VALIDAR DIVISA

INICIO

OBTEN divisa

SI divisa = "dólar" ENVIA (al proceso 3.7) divisa-válida

SINO ENVIA (al operador) respuesta-cambio-configuración

= "Error Divisa Inválida: La divisa actual es el dólar"

FIN SI

FIN

Proceso 3.7 ACTUALIZAR CONFIGURACION

INICIO

ENVIA (al FEC) configuración-válida

RECIBE respuesta-configuración DEL FEC

SI respuesta-configuración = éxito

ENVIA (al operador) respuesta-cambio-configuración = "La configuración ha sido cambiada"

SINO

ENVIA (al operador) respuesta-cambio-configuración

= "Error: La configuración no pudo ser cambiada"

FIN SI

FIN

Proceso 5. ACTUALIZAR ESTADO BANCO

INICIO

RECIBE estado-conexión-banco DEL FEC
PON nuevo-estado COMO estado-conexión-banco
PON estado-conexión-banco COMO nuevo-estado
ENVIA (al operador) estado-conexión-banco

FIN

Proceso 6. DISTRIBUIR INDICACIONES

INICIO

RECIBE datos-indicaciones DEL operador
SI SELECCIONO MENSAJE
CONSTRUYE MENSAJE CON texto
ENVIA (al FEC) bancos, mensaje
SI NO CONSTRUYE MARQUESINA CON texto
ENVIA (al FEC) bancos, marquesina
FIN SI

FIN

Proceso 7.1. VERIFICAR NUMERO SELECCIONADOS

INICIO

RECIBE solicitud-inicio-subasta DEL operador
SI participantes núm-participantes
ENVIA (al operador) "Error: El mínimo número de
participantes necesarios
para ejecutar la subasta es núm-participantes"
TERMINA PROCESO
FINSI
ENVIA (al FEC) solicitud-inicio-subasta
SI RECIBE inicia-subasta DEL FEC
ENVIA (al operador) inicia-subasta
FIN SI

FIN

Proceso 8. ACTUALIZAR GRAFICA POSTURAS

INICIO

RECIBE posturas DEL FEC
CON posturas CREA gráfica-posturas
ENVIA (al operador) gráfica-posturas

FIN

Proceso 9. ACTUALIZAR TABLA ARBITRAJES

INICIO

RECIBE arbitrajes DEL FEC
CON arbitrajes CREA tabla-arbitrajes
ENVIA (al operador) tabla-arbitrajes

FIN

Proceso 10. REALIZAR ASIGNACION

INICIO

RECIBE arbitraje-asignación DEL operador

OBTEN precio-compra DE arbitraje-asignación

OBTEN precio-venta DE arbitraje-asignación

SI precio-compra > precio-venta

ENVIA (al FEC) arbitraje-asignación-válida

SINO

ENVIA (al operador) respuesta-asignación = "Error:

La asignación es

inválida"

FINSI

RECIBE respuesta-asignación DEL FEC

SI respuesta-asignación = éxito

ENVIA (al operador) respuesta-asignación = "Exito:

La asignación fué

aceptada"

SINO

ENVIA (al operador) respuesta-asignación = "Error: La

asignación fué

rechazada"

FIN SI

FIN

Proceso 11. INFORMAR TIEMPO SESION

INICIO

RECIBE tiempo-sesión DEL FEC

CON tiempo-sesión ACTUALIZA reloj

MUESTRA (al operador) tiempo-sesión

FIN

Proceso 12. INFORMAR TIEMPO POSTURAS TERMINADO

INICIO

RECIBE fin-posturas del FEC

ENVIA fin-posturas AL operador

FIN

Proceso 13. INFORMAR INICIO ARBITRAJES

INICIO

RECIBE inicio-arbitrajes del FEC

ENVIA fin-arbitrajes AL operador

FIN

Proceso 14. INFORMAR TIEMPO ARBITRAJES TERMINADO

INICIO

RECIBE fin-arbitrajes del FEC

ENVIA fin-arbitrajes AL operador

FIN

Proceso 15. INFORMAR SESIÓN ESPERANDO CLIENTES
INICIO

RECIBE sesión-esperando-clientes del FEC
ENVIA sesión-esperando-clientes AL operador

FIN

Proceso 16. INFORMAR SESIÓN FINALIZADA
INICIO

RECIBE fin-sesión del FEC
ENVIA fin-sesión AL operador

FIN

Proceso 17. MOSTRAR RESULTADOS FIX-SESIÓN
INICIO

RECIBE FIX-sesión del FEC
MUESTRA FIX-sesión AL operador

FIN

Proceso 18. MOSTRAR RESULTADOS FIX DIA
INICIO

RECIBE resultados-FIX del FEC
OBTEN FIX-día DE resultados-FIX
MUESTRA FIX-día AL operador
OBTEN TCVM-día DE resultados-FIX
MUESTRA TCVM-día AL operador

FIN

Proceso 19. OBTENER HISTORICO SESION
INICIO

OBTEN núm-histórico DEL operador
CON núm-histórico SOLICITA datos-histórico AL FEC
RECIBE datos-histórico DEL FEC
CON datos-histórico CREA histórico-sesión
MUESTRA histórico-sesión AL operador

FIN

Proceso 20. OBTENER ASIGNACIONES
INICIO

RECIBE (del operador) solicitud-resultado-asignaciones
ENVIA solicitud-resultado-asignaciones AL FEC
RECIBE asignaciones DEL FEC
CON asignaciones CREA reporte-asignaciones
MUESTRA reporte-asignaciones AL operador

FIN

Proceso 21. PUBLICAR FIX-DIA INTERNET

INICIO

RECIBE solicitud-publicar-Internet DEL operador
ENVIA (al FEC) solicitud-publicar-Internet
RECIBE respuesta-publicar-Internet DEL FEC
ENVIA (al operador) respuesta-publicar-Internet

FIN

Proceso 22.1 SELECCIONAR CARTA

INICIO

OBTEN solicitud-cartas DEL operador
OBTEN selección DEL operador
SI selección = selección -comunicación-social
ENVIA (al proceso 16.2) selección-comunicación-social
SINO

ENVIA (al proceso 16.3) selección-comunicado FINSI

FIN

Proceso 22.2 GENERAR CARTA COM-SOC

INICIO

OBTEN formato-com-soc DE FORMATO COMUNICACION SOCIAL
ACTUALIZA resultados-FIX EN formato-com-soc
PON carta-com-soc COMO formato-com-soc
ENVIA (al operador) carta-com-soc

FIN

Proceso 22.3 GENERAR CARTA COMUNICADO

INICIO

OBTEN formato-comunicado DE FORMATO COMUNICADO
ACTUALIZA resultados-FIX EN formato-comunicado
PON carta-comunicado COMO formato-comunicado
ENVIA (al operador) carta-comunicado

FIN

Proceso 23.1 SELECCIONAR IMPRESION

INICIO

OBTEN solicitud-imprimir DEL operador
OBTEN selección DEL operador
HACER CASO
CAS01 selección = selección -histórico
ENVIA (al proceso 17.2) histórico
CAS02 selección = selección-asignaciones
ENVIA (al proceso 17.3) asignaciones
CAS03 selección = selección-comunicación-social
ENVIA (al proceso 17.4) carta-comunicación-social
CAS04 selección = selección-comunicado

ENVIA (al proceso 17.5) carta-comunicado TERMINA CASO
FIN

Proceso 23.2 IMPRIMIR HISTORICO

INICIO

RECIBE histórico

ENVIA (a la impresora) histórico

RECIBE respuesta-impresión DE LA impresora

ENVIA (al operador) respuesta-impresión

FIN

Proceso 23.3 IMPRIMIR ASIGNACIONES

INICIO

RECIBE asignaciones

ENVIA (a la impresora) asignaciones

RECIBE respuesta-impresión DE LA impresora

ENVIA (al operador) respuesta-impresión

FIN

Proceso 23.4 IMPRIMIR CARTA COM-SOCIAL

INICIO

RECIBE carta-comunicación-social

ENVIA (a la impresora) carta-comunicación-social

RECIBE respuesta-impresión DE LA impresora

ENVIA (al operador) respuesta-impresión

FIN

Proceso 23.5 IMPRIMIR CARTA COMUNICADO

INICIO

RECIBE carta-comunicado

ENVIA (a la impresora) carta-comunicado

RECIBE respuesta-impresión DE LA impresora

ENVIA (al operador) respuesta-impresión

FIN

Proceso 24.1 SELECCIONAR DOCUMENTO

INICIO

OBTEN tipo-documento, nombre-archivo DEL operador

OBTEN selección DEL operador

HACER CASO

CAS01 selección = selección-histórico

ENVIA (al proceso 18.2) histórico, nombre-archivo

CAS02 selección = selección-asignaciones

ENVIA (al proceso 18.3) asignaciones, nombre-archivo

CAS03 selección = selección-comunicación-social

ENVIA (al proceso 18.4) carta-comunicación-social,
nombre-archivo

CASO4 selección = selección-comunicado

ENVIA (al proceso 18.5) carta-comunicado, nombre-
archivo

TERMINA CASO

FIN

Proceso 24.2 GUARDAR HISTORICO

INICIO

RECIBE histórico, nombre-archivo

CREA HISTORICOS CON nombre-archivo

GUARDA histórico EN HISTORICOS

ENVIA (al operador) respuesta-guardar

FIN

Proceso 24.3 GUARDAR ASIGNACIONES

INICIO

RECIBE asignaciones, nombre-archivo

CREA ASIGNACIONES CON nombre-archivo

GUARDA asignaciones EN ASIGNACIONES

ENVIA (al operador) respuesta-guardar

FIN

Proceso 24.4 GUARDAR CARTA COM-SOCIAL

INICIO

RECIBE carta-comunicación-social, nombre-archivo

CREA COMUNICACION-SOCIAL CON nombre-archivo

GUARDA carta-comunicación-social EN COMUNICACION-SOCIAL

ENVIA (al operador) respuesta-guardar

FIN

Proceso 24.5 GUARDAR CARTA COMUNICADO

INICIO

RECIBE carta-comunicado, nombre-archivo

CREA COMUNICADOS CON nombre-archivo

GUARDA carta-comunicado EN COMUNICADOS

ENVIA (al operador) respuesta-guardar

FIN

Proceso 25. SALIR APLICACIÓN

INICIO

SI RECIBE servidor-no-disponible DEL FEC

ENVIA id-error= servidor-no-disponible AL operador

SI RECIBE desconexión DEL FEC

ENVIA id-error= desconexión AL operador

FIN SI

ENVIA fin-programa AL operador

FIN

Proceso 26. ENVIAR MENSAJE "VIVO"

INICIO

SI RECIBE MENSAJE estas-vivo DEL FEC

ENVIA MENSAJE vivo AL FEC

FIN SI

FIN

El diagrama final para completar la especificación estructurada es el diagrama de transición de estados, el cual se presenta en la figura 2.7.

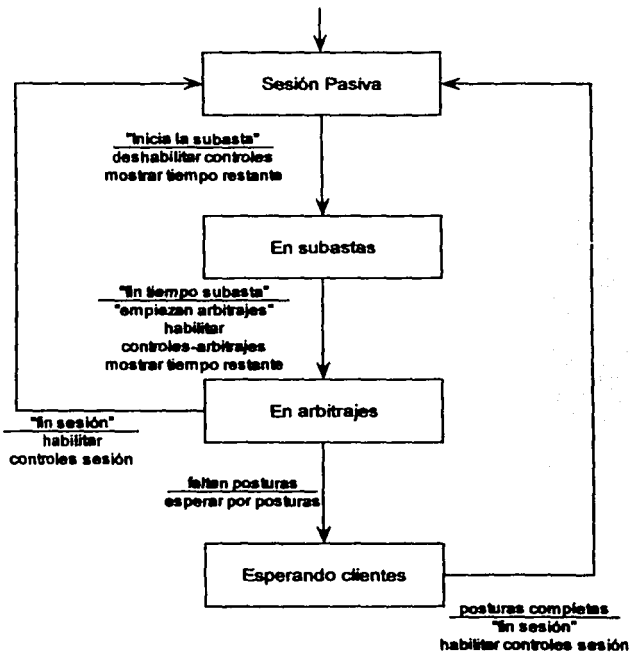


Figura 2.7 Diagrama de transición de estados.

En éste diagrama se muestra que el sistema puede alcanzar cuatro estados. El estado en *sesión pasiva* se caracteriza porque el operador puede hacer todas aquellas actividades previas o siguientes a una subasta, tales como: cambiar la configuración, seleccionar participantes, imprimir resultados, acceder a históricos.

Una vez que se inicia una subasta el sistema entra en el estado de *subastas* el cual tiene una duración establecida de 70 segundos, todos los controles deben ser desactivados para que nada interfiera con el desarrollo de la subasta.

Al finalizar el tiempo de la subasta el sistema entra en el estado de *arbitrajes* el cual dura diez segundos. En este estado el operador solo puede seleccionar arbitrajes, es decir, solo algunos controles podrán estar activos.

Después del tiempo de arbitrajes, el sistema puede regresar al estado de *sesión pasiva* solo si se recibieron todas las posturas, con las cuales se calcula el resultado del FLX y se da por terminada la sesión. En caso de no recibir todas las posturas el sistema pasará a un estado de *espera de clientes* en el que permanecerá hasta que la última postura requerida haya sido recibida.

Para terminar el modelo esencial del sistema se completó el diccionario de datos (apéndice A) con la definición de los componentes de los diagramas de flujo de datos, miniespecificaciones, diagramas de transición de estados y demás modelos empleados en la especificación estructurada.

Capítulo 3

DISEÑO

En este capítulo se presentarán los modelos de Implementación del Usuario, de Implementación del Sistema y de Implementación del Programa, como parte de la metodología de diseño estructurado, y los cuáles fueron la base para el desarrollo del sistema.

3.1 El modelo de implementación del usuario para FixBan cliente Bancico

El modelo de implementación del usuario es el punto intermedio entre el análisis y el diseño. Determina los límites de automatización del sistema, los requerimientos de entrada / salida, interfaz gráfica del usuario y limitaciones operacionales.

3.1.1 Límites de automatización

Los límites de automatización establecen la frontera entre el sistema y el usuario, es decir, indican donde termina la operación del sistema y empieza la del usuario. Los límites de automatización se especifican en los diagramas de flujo de datos de bajo nivel. Particularmente las tareas cubiertas por el usuario son: tecleo de datos, tales como: el login, password y datos de la configuración, selección de operaciones, tales como: ejecución de la subasta e impresión de resultados, y toma de decisión, tal como: la realización de arbitrajes. La figura 3.1 muestra los límites de automatización del sistema FixBan cliente Bancico.

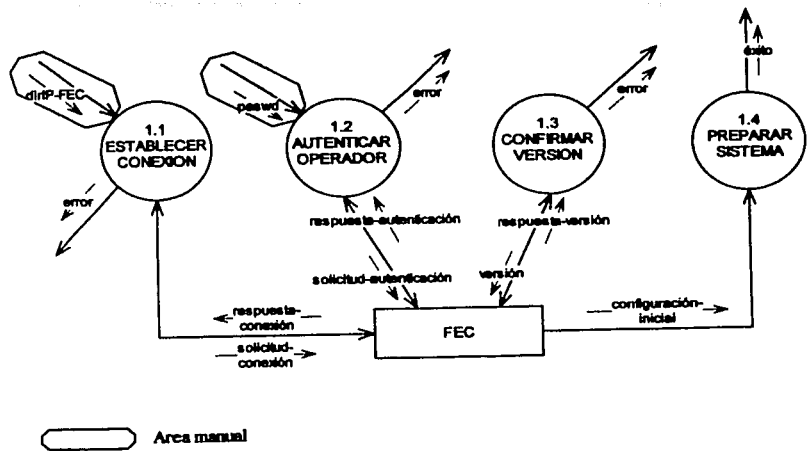


Figura 3.1 Límites de Automatización. Proceso 1 Entrar al Sistema.

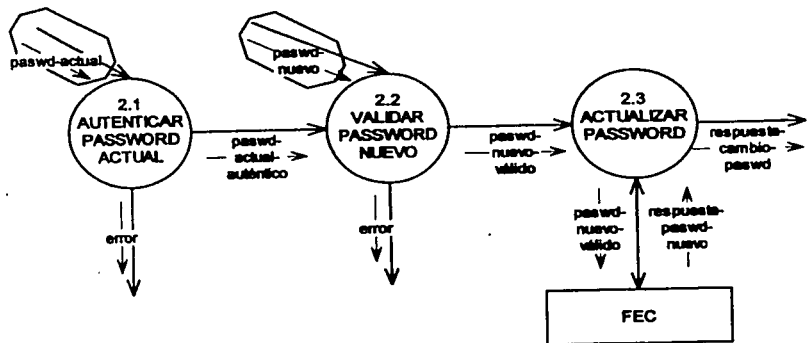


Figura 3.1 Límites de Automatización (continuación). Proceso 2 Cambiar Password.

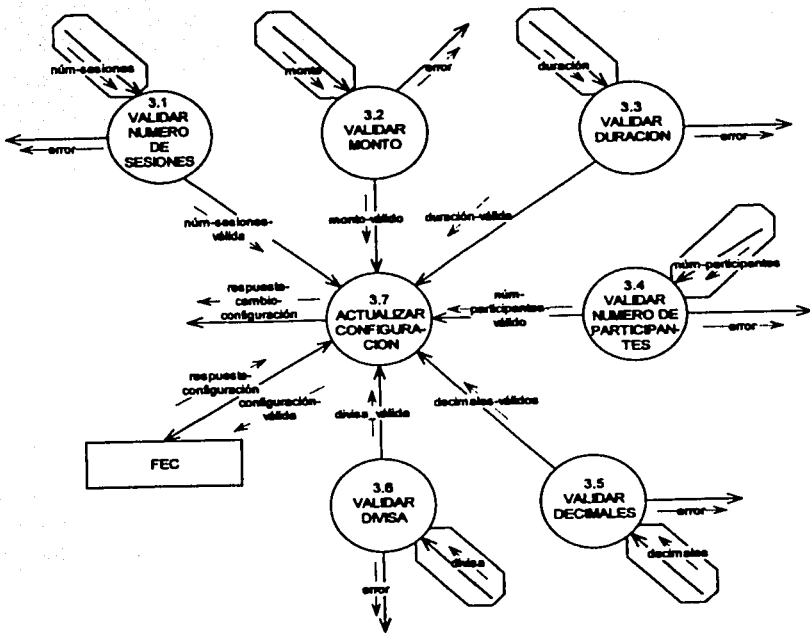


Figura 3.1 Límites de Automatización (continuación). Proceso 3 Cambiar Configuración.

TESIS CON
FALLA DE ORIGEN

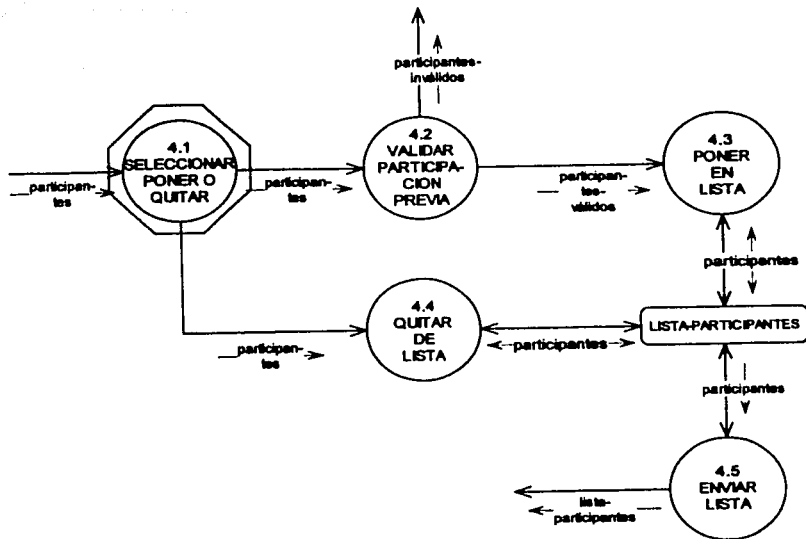


Figura 3.1 Límites de Automatización (continuación). Proceso 4 Actualizar Lista de Participantes.

TESIS CON
FALLA DE ORIGEN

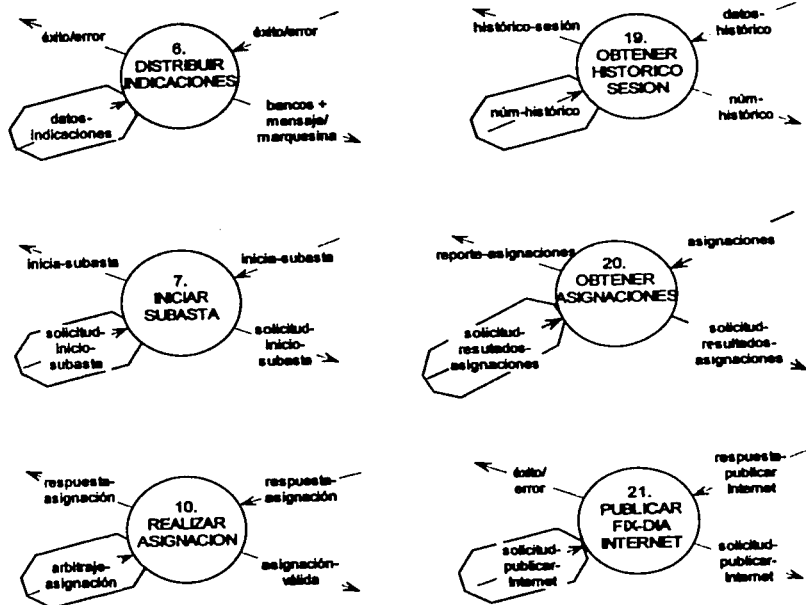


Figura 3.1 Límites de Automatización (continuación) Procesos 6, 7, 10, 19, 20 y 21.

TESIS CON
FALLA DE ORIGEN

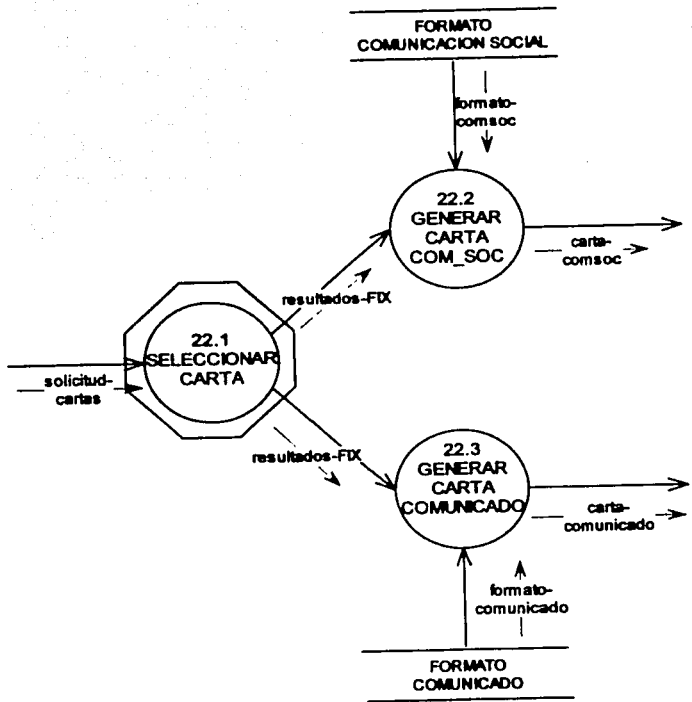


Figura 3.1 Limites de Automatización (continuación). Proceso 22 Generar Cartas.

TESIS CON
FALLA DE ORIGEN

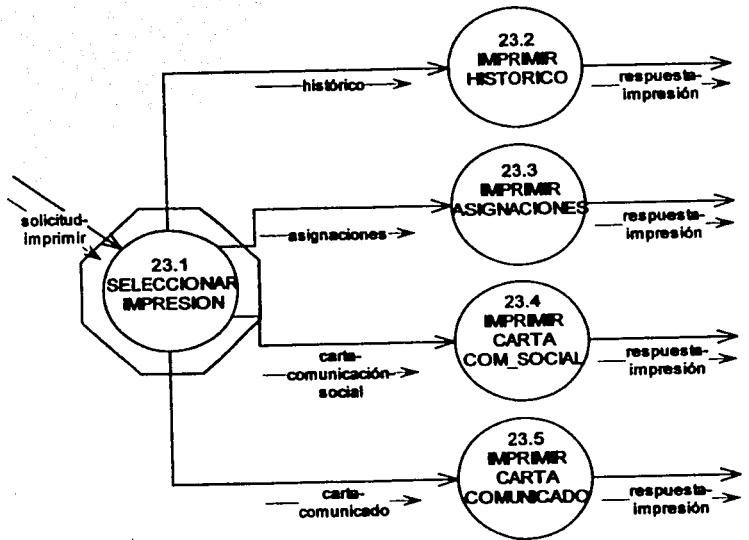


Figura 3.1 Límites de Automatización (continuación). Proceso 23 Imprimir.

TESIS CON
FALLA DE ORIGEN

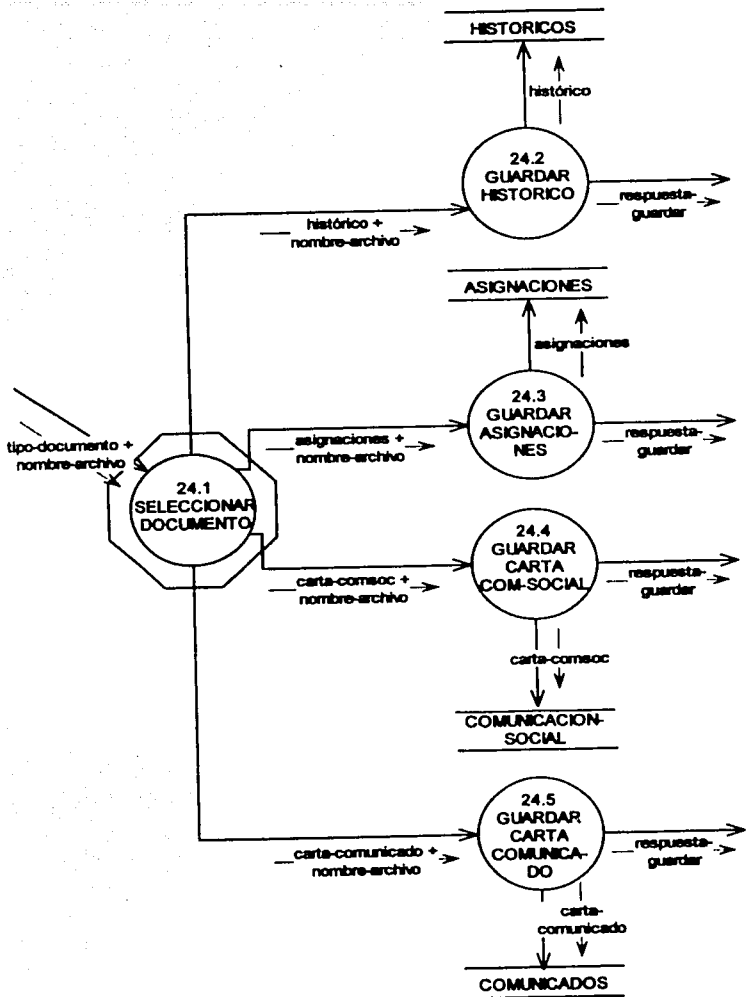


Figura 3.1 Límites de Automatización (continuación). Proceso 24 Guardar.

3.1.2 Formatos de entrada y salida

Para la especificación de los formatos de salida se siguió con el estilo usado en el sistema previo, debido a la simplicidad y facilidad de uso.

La figura 3.2 muestra los formatos propuestos para la impresión de los reportes.



Figura 3.2 Formatos de salida. Carta para Comunicación Social.



BANCO DE MÉXICO

TIPO DE CAMBIO PARA SOLVENTAR OBLIGACIONES DENOMINADAS EN MONEDA EXTRANJERA PAGADERAS EN LA REPUBLICA MEXICANA

Con fundamento en el artículo 35 de la Ley del Banco de México; en los artículos 8o. Y 10o. del Reglamento Interior del Banco de México, y en los términos del numeral 1.2 de las Disposiciones Aplicables a la Determinación del Tipo de Cambio para Solventar Obligaciones Denominadas en Moneda Extranjera Pagaderas en la República Mexicana, publicadas en el Diario Oficial de la Federación el [fecha], el Banco de México informa que el tipo de cambio citado obtenido el día de hoy conforme al procedimiento establecido en el numeral 1 de las Disposiciones mencionadas, fue de [fix] M.N. [fix_letra] por un dólar de los EE.UU.A.

La equivalencia del peso mexicano con otras monedas extranjeras se calculará atendiendo a la cotización que rija para estas últimas contra el dólar de los EE.UU.A., en los mercados internacionales el día en que se haga el pago. Estas cotizaciones serán dadas a conocer, a solicitud de los interesados, por las instituciones de crédito del país.

Atentamente,

BANCO DE MEXICO

MEXICO, D.F., A [fecha2].

Figura 3.2 Formatos de salida (continuación). Comunicado.

FixBan [version]



**REGISTROS DE COTIZACIONES
PROPORCIONADAS POR LA BANCA**

Sesión: [sesion]

Fecha: [fecha]

CLAVE	INSTITUCION	T.COMPRAS	T.VENTA
cve-a	Institución-a	compra-1	venta-1
cve-b	Institución-b	compra-2	venta-2
cve-c	Institución-c	compra-3	venta-3
cve-d	Institución-d	compra-4	venta-4

TIPO DE CAMBIO DE LA SESION: [fix]

Figura 3.2 Formatos de salida (continuación). Cotizaciones finales.

TESIS CON
FALLA DE ORIGEN



BANCO DE MEXICO

FixBan [version]
P O S T U R A S

Sesión: [sesion]

Fecha: [fecha]

CLAVE	INSTITUCION	PRECIO COMPRA	PRECIO VENTA	HORA	ID
cve-a	Institución-a	compra-1	venta-1	h-1	a
cve-a	Institución-a	compra-2	venta-2	h-2	a
cve-b	Institución-b	compra-1	venta-1	h-1	b
cve-b	Institución-b	compra-2	venta-2	h-2	b
cve-b	Institución-b	compra-3	venta-3	h-3	b
cve-b	Institución-b	compra-4	venta-4	h-4	b
cve-c	Institución-c	compra-1	venta-1	h-1	c
cve-d	Institución-d	compra-1	venta-1	h-1	d

TIPO DE CAMBIO DE LA SESION: [fix]

Figura 3.2 Formatos de salida (continuación). Todas las cotizaciones.



BANCO DE MEXICO

FixBan [version]
ARBITRAJES

Sesión: [sesion]

Fecha: [fecha]

BANCO COMPRA	PRECIO COMPRA	PRECIO VENTA	BANCO VENTA
Institución-1	compra-1	venta-1	Institución-2
Institución-1	compra-1	venta-2	Institución-3
Institución-1	compra-1	venta-3	Institución-4

ASIGNACIONES

BANCO COMPRA	PRECIO COMPRA	PRECIO VENTA	BANCO VENTA
Institución-1	compra-1	venta-1	Institución-2

Figura 3.2 Formatos de salida (continuación). Arbitrajes y asignaciones.

FALTA

PÁGINA

74 |

3.1.3 Diseño de formas

Se desarrolló un diseño preliminar de las pantallas de cada módulo. La figura 3.3 muestra los diseños resultantes.

The image shows a graphical user interface window titled "FibBar 2.0: Entrada al sistema". Inside the window, there is a central rectangular form. The form contains three labeled input fields: "Servidor:" followed by a text box with a small downward-pointing arrow icon; "Login:" followed by a text box containing the text "BANXICO"; and "Password:" followed by an empty text box. Below the form, there are two rounded rectangular buttons: "Aceptar" on the left and "Cancelar" on the right.

Figura 3.3 Diseño del GUI. Módulo de entrada al sistema.

TESIS CON
FALLA DE ORIGEN

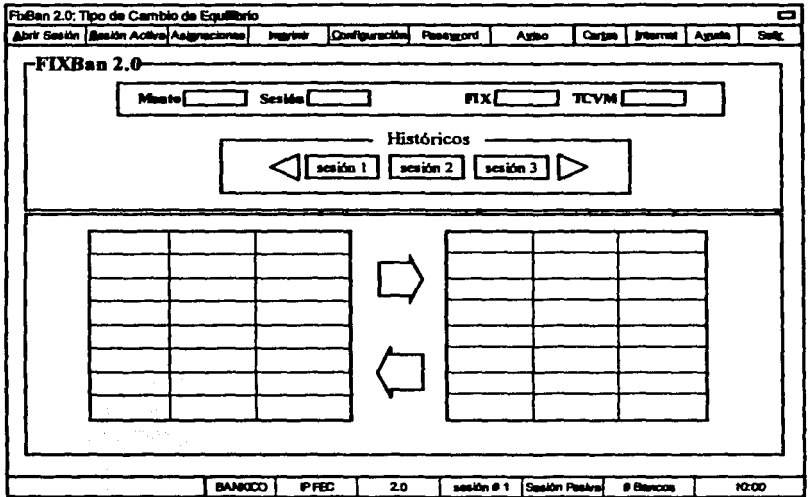


Figura 3.3 Diseño del GUI (continuación). Módulo principal.

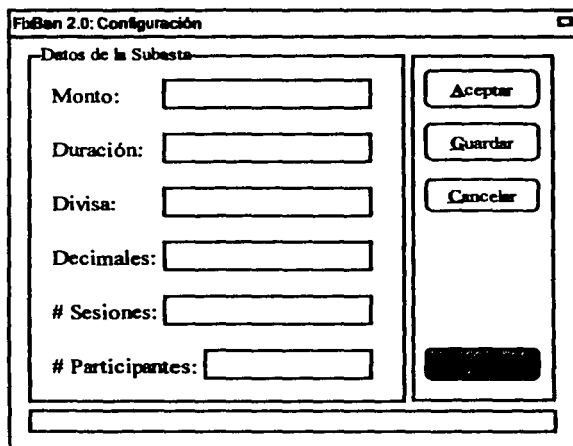


Figura 3.3 Diseño del GUI (continuación). Módulo configuración.

TESIS CON
FALLA DE ORIGEN

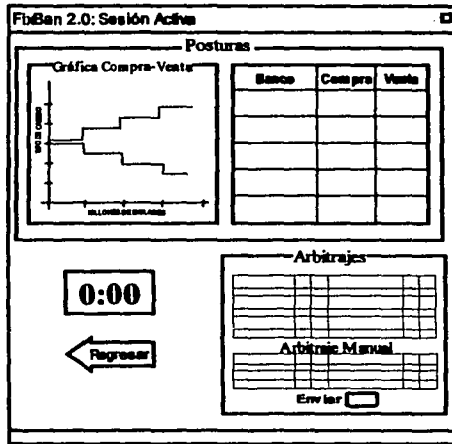


Figura 3.3 Diseño del GUI (continuación). Módulo subastas.

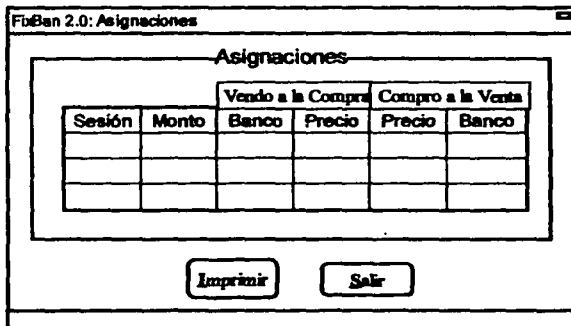


Figura 3.3 Diseño del GUI (continuación). Módulo asignaciones.

FxBan 2.0: Históricos □

Configuración Posturas Arbitrajes Operaciones Imprimir Salir

Sesión #1 Banco de Mexico 01/01/01
 FxBan 2.0

Tipo de Cambio de Equilibrio

Fix Sesión: TCVM: FIX Día:

Configuración

Monto: Divisa: # Sesiones:
 Duración: Decimales: # Participantes:

Posturas

Gráfica Compra-Venta

Banco	Compra	Venta	Hora	ID

Arbitrajes

No hubo arbitrajes

Operaciones

No hubo operaciones

Figura 3.3 Diseño del GUI (continuación). Módulo de históricos.

TESIS CON
FALLA DE ORIGEN

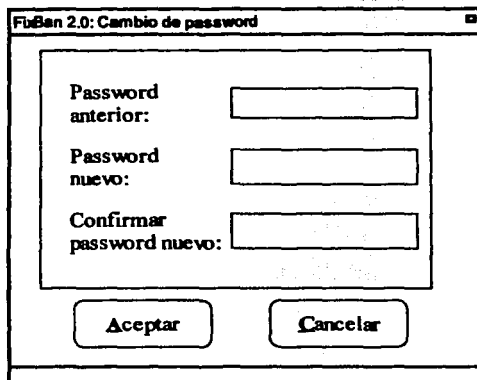


Figura 3.3 Diseño del GUI (continuación). Módulo cambio de password.

TESIS CON
FALLA DE ORIGEN

FinBan 2.0: Aviso a todos

Mensaje:

Marguesina:

Enviar a: Todos Selección

Atlantic	BSV	Citibank	Inverlat	Mifel
Banamax	Bibaf	Dreschner	Inveex	Nefin
Bancomer	BNP	Inbursa	Ice	RNB
Bancrocor	Bofa	Industrial	JPMorgan	Serfin
Banorte	Chase	ING	Mexicano	

Figura 3.3 Diseño del GUI (continuación). Módulo distribución de indicaciones.

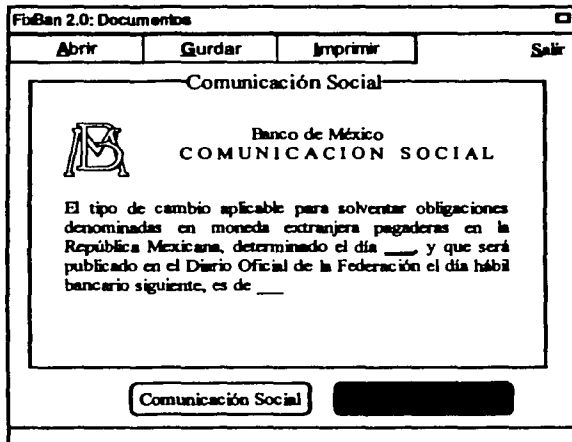


Figura 3.3 Diseño del GUI (continuación). Módulo cartas.

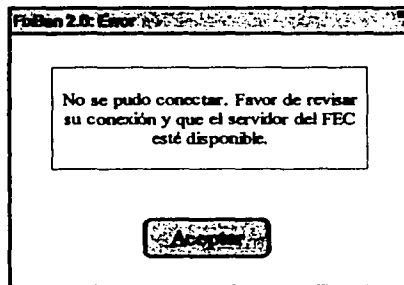


Figura 3.3 Diseño del GUI (continuación). Pantalla de error.

3.1.4 Límites operacionales.

Las limitaciones operacionales son requerimientos que el sistema debe satisfacer tales como volumen de datos, tiempos de respuesta, limitaciones políticas y ambientales, de seguridad y robustez, y que ayudan a delimitar el espacio solución del diseño.

3.1.4.1 Volumen de datos.

Una limitante para el manejo de datos fue el uso del protocolo de comunicación FEC [Hernández 1999]. En este protocolo los datos se organizan en mensajes que constan de un encabezado y un cuerpo, y se construyen con los tipos de datos: char, int, short, y string.

La longitud del encabezado es de 4 bytes. El primer byte especifica el destinatario, el siguiente el tipo de mensaje (formato) y los últimos dos el tamaño del mensaje. Por otra parte, la longitud del cuerpo del mensaje es variable.

Cada mensaje tiene un formato previamente establecido y un identificador asignado. Este se construye con expresiones regulares formadas con los tipos de datos básicos mencionados previamente. El tipo de dato char es representado con %c y tiene un tamaño de 1 byte, int se representa con %l y mide 4 bytes, short se representa a través de %d y su longitud es de 2 bytes, string se representa con %s.

Para interpretar los mensajes se obtiene del encabezado el tipo de mensaje, por ejemplo, el mensaje 1 está reservado para LOGIN, y su formato es %s, de tal manera que en el cuerpo del mensaje se obtendrá una cadena.

La tabla 3.1 contiene la lista de los formatos creados para los mensajes entre el cliente Banxico y el FEC.

Mensaje	Id	Formato
LOGOUT	0	%c%c%d
LOGIN	1	%s
PSSWD	2	%s
IMALIVE	7	%c%c%d
INICIATALK	16	%c%c%d
DENTROSESION	13	%c%c%d
MANDAVERSION	31	%d%d
DAPERMISO	32	%n%d
QUITAPERMISO	33	%n%d
INICIARSESION	34	%d%d%c%d%d
PIDELISTABANCOS	36	%c%c%d
NUEVACONFIGURACION	37	%d%d%d%c%d%d%d
PIDEFIXDIARIO	38	%c%c%d
SELECCIONAARBITRAJE	39	%d%d%d%l
PIDEASIGNACIONES	40	%c%c%d
PIDEINFOSESION	41	%d
MANDAMSG	43	%n%s
MANDAMSGCINTILLA	44	%s

PUBLICAHECHOSINTERNET	45	%c%c%d
MSGALGUNOS	46	%n(%d)%n(%s)
CINTILLAALGUNOS	47	%n(%d)%s
PIDEINFCORTO	48	%d
MANDAPASSWORD	49	%s
CHANGEPSWD	212	%s
LOGINREQ	254	%c%c%d
PSSWDREQ	255	%c%c%d
PIDEPASSWORD	125	%c%c%d
VERSIONINCORRECTA	128	%c%c%d
VERSIONCORRECTA	127	%c%c%d
LISTABANCOS	128	%n%d%c%s%s%s%d
PRINCIPIASESION	129	%d%d%c%d%d
TIEMPOESTANTE	130	%d
TIEMPOARBITRAJES	131	%d
POSTURARECIBIDA	132	%d%d%l
FINSESION	134	%c%c%d
LLEGABANCO	138	%d
SEVABANCO	139	%d
CONFIGURACIONINICIAL	140	%d%d%d%c%d%d%d%d
CONFIGURACIONACEPTADA	141	%c%c%d
CONFIGURACIONRECHAZADA	142	%c%c%d
COMIENZAARBITRAJE	143	%c%c%d
FINARBITRAJE	144	%c%c%d
FINTIEMPOPARTICIPACION	145	%c%c%d
FIXSESION	146	%d%d%l
FIXDIARIO	147	%d%l%d%l
ARBITRAJESPOSTURAS	148	%d%d%n(%d%l%d%l)
RECIBEFIXES	150	%n(%d%l)
POSTURASVIGENTES	151	%d%n(%d%l%l)
ARBITRAJEACEPTADO	153	%c%c%d
ARBITRAJERECHAZADO	154	%c%c%d
ASIGNACIONES	156	%n(%d%d%d%d%l%l%l)
INFOSESION	157	%d%d%d%d%c%d%l%l%d%n(%d%l%d%l)%n(%d%l%l%l)
VALORESANTERIORES	159	%n(%d%l)
RECIBEMSG	160	%n%s
RECIBECINTILLA	181	%s
INTERNETHECHO	182	%d
INFOSESIONCORTO	185	%d%l%d%n(%d%l%l)
CONFPASSWD	236	%c%c%d
DEADSRVR	243	%c%c%d
AREYOUALIVE	245	%c%c%d
PSSWDFAIL	249	%c%c%d
LOGINFAIL	251	%c%c%d
SERVFUERADESERVICIO	252	%c%c%d
LOGGED	253	%d

Tabla 3.1 Formatos del protocolo del FEC usados por el cliente Bancico.

TESIS CON
FALLA DE ORIGEN

3.1.4.2 Tiempo de respuesta

Se identificaron las siguientes situaciones donde el sistema debe responder en un tiempo especificado:

- Para establecer conexión con el FEC el sistema debe dejar pasar al menos 60 segundos para recibir respuesta antes de considerar la conexión fallida.
- El sistema tiene que responder a los mensajes AREYOUALIVE del FEC con un mensaje IMALIVE. Si el FEC no recibe el mensaje de contestación el cliente será desconectado.
- El sistema tiene 70 segundos (o según la configuración) para mostrar el desarrollo de la subasta, y 10 segundos para responder con arbitrajes.

3.1.4.3 Limitaciones políticas

La primera limitación fue establecida por la subgerencia encargada de mantener el funcionamiento del servicio del FIX, el resto se especifican en el Diario Oficial de la Nación [Diario Oficial 1996].

- El sistema deberá ser una aplicación de 32-bits para Windows. El lenguaje de programación especificado fue Visual Basic 5.0.
- El mínimo número de participantes en la subasta es de 4.
- El mínimo número de sesiones en el día es 3.
- Los participantes solo podrán participar una sola vez en el día.

3.1.4.4 Seguridad

Las restricciones de seguridad fueron propuestas por nosotros.

- Se debe proveer autenticación del operador a través de password. Las restricciones para la elección del password son las siguientes: combinación de letras mayúsculas o minúsculas, con números o caracteres especiales.
- Se le permitirán al operador hasta 3 intentos para introducir el password correcto antes de negarle el acceso al sistema.

3.1.4.5 Disponibilidad

Estas restricciones son de interés primordial, ya que del funcionamiento diario del sistema depende el mercado de dólares en México.

- Es de interés primordial que el sistema funcione todos los días hábiles para ejecutar las subastas en lapsos de 15 minutos que podrán ocurrir de las 9:00 horas a las 9:59, de las 10:00 a las 10:59 y de las 11:00 a las 12:00.

3.2 El modelo de implementación del sistema para FixBan cliente Banxico

El modelo de implementación del sistema consiste de dos modelos: el modelo de procesadores y el modelo de tareas.

3.2.1 El modelo de procesadores

Para el modelo de procesadores del sistema completo FixBan 2.0 se escogió como solución el uso de varias arquitecturas de computadoras. La figura 3.4 muestra que el cliente Banxico está alojado en una computadora personal, el FEC se corre en una estación de trabajo Sun y el servidor se corre en una segunda estación de trabajo Sun.

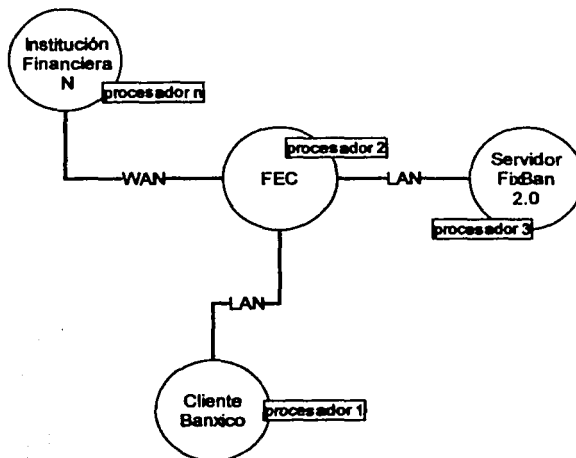


Figura 3.4 Modelo de procesadores.

Particularmente el cliente Banxico solo emplea el uso de un procesador.

3.2.2 El modelo de tareas

Consiste en organizar cada procesador en áreas individuales o tareas. La tabla 3.2 muestra el modelo de tareas del cliente Banxico del sistema FixBan.

Se identificaron tres tareas: la de comunicaciones, la de atención al operador y la de atención al FEC.

La tarea de comunicaciones se encarga de mantener activa la comunicación con el FEC, es decir, responder con un mensaje "vivo" a los mensajes "estas vivo" enviados por el FEC y atender los mensajes de desconexión.

La tarea de atención al operador responde a los comandos del operador, es decir, se encarga de que se lleven a cabo todas las peticiones hechas por el operador.

La tarea de atención al FEC maneja los mensajes provenientes del servidor a través del FEC, algunos de los cuáles incluyen las respuestas a las peticiones hechas por el operador.

Sistema Operativo		
Comunicaciones	Atención Operador	Atención FEC

Tabla 3.2 Modelo de tareas.

3.3 Modelo de implementación del programa para FixBan cliente Banxico

En el modelo de implementación del programa se desarrollaron diagramas de estructuras para mostrar los módulos en los que se dividen cada tarea. Es importante indicar que el objetivo de éstos diagramas es mostrar la jerarquía de los módulos sin enfatizar el flujo de datos o la secuencia de los módulos. La figura 3.5 muestra los diagramas de estructuras realizados.

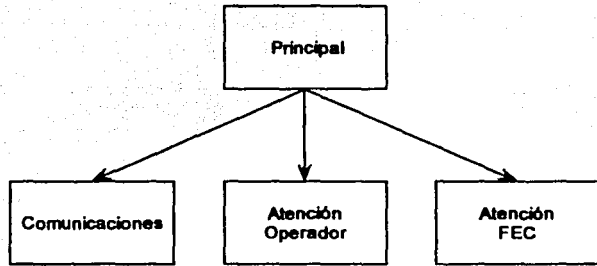


Figura 3.5 Diagrama de estructuras. Módulos principales del sistema.

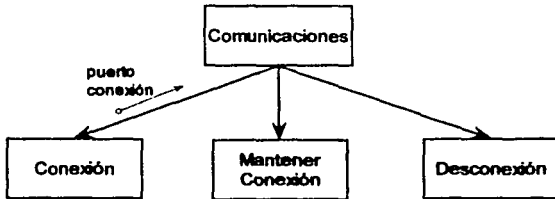


Figura 3.5 Diagrama de estructuras (continuación). División del módulo de comunicaciones.

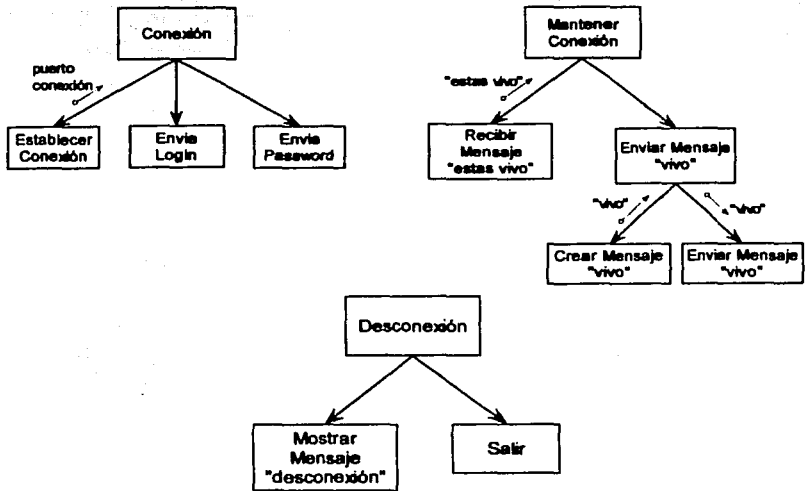


Figura 3.5 Diagrama de estructuras. División del módulo de comunicaciones (continuación).

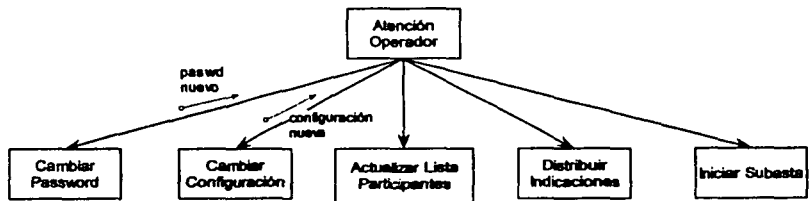


Figura 3.5 Diagrama de estructuras (continuación). División del módulo de Atención al operador.

TESIS CON
FALLA DE ORIGEN

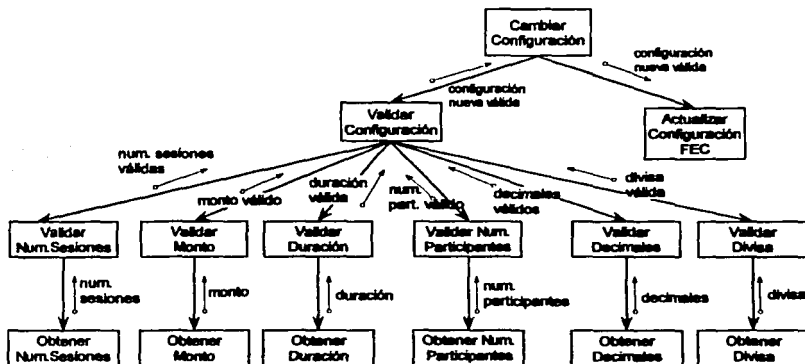
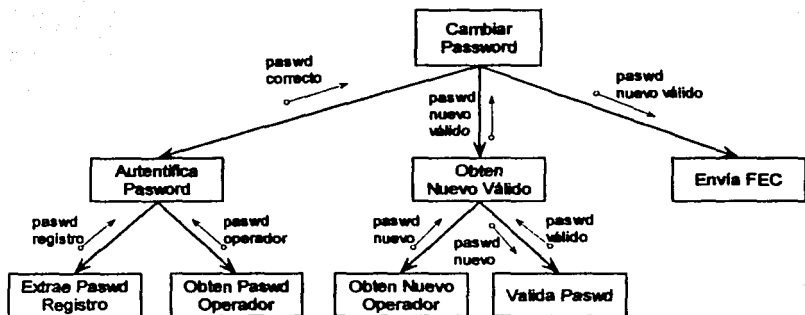


Figura 3.5 Diagrama de estructuras. División del módulo de Atención al operador (continuación).

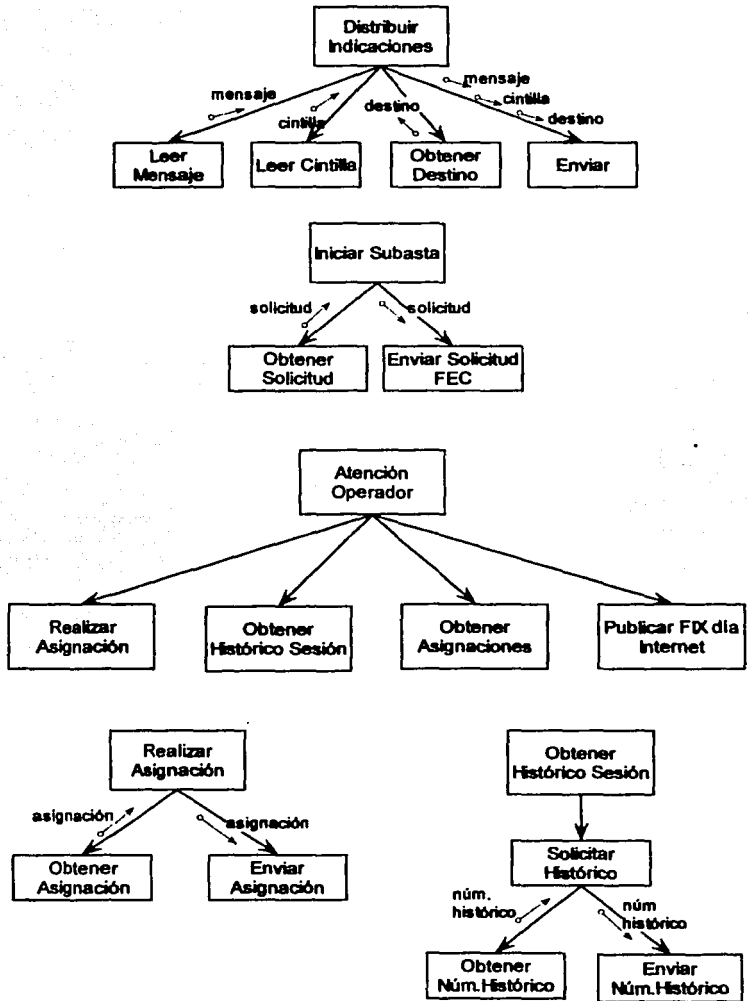


Figura 3.5 Diagrama de estructuras. División del módulo de Atención al operador (continuación).

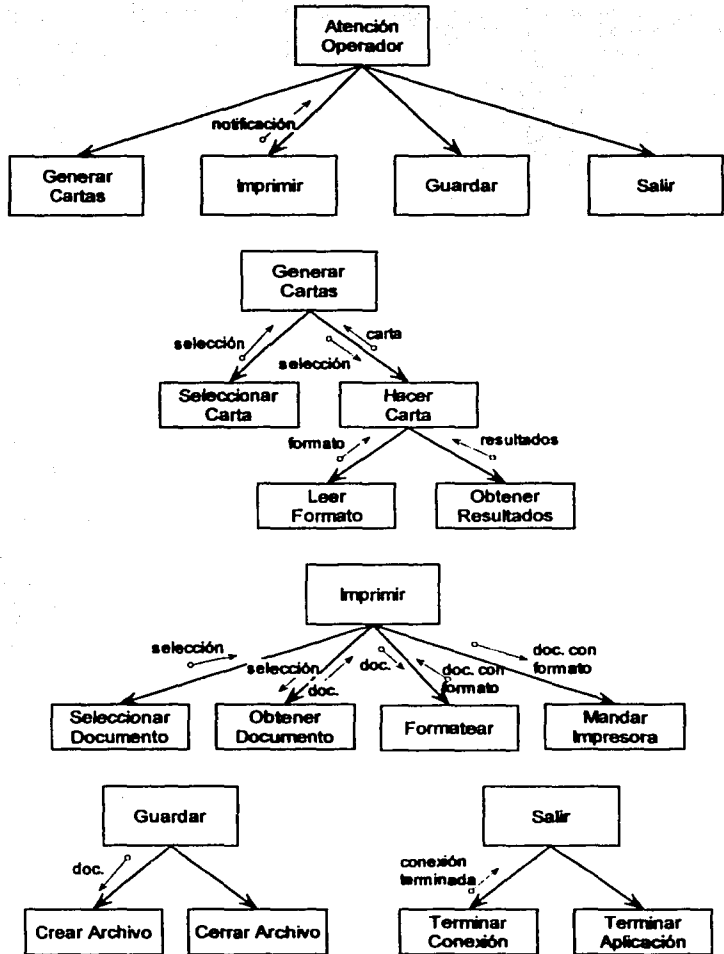


Figura 3.5 Diagrama de estructuras. División del módulo de Atención al operador (continuación).

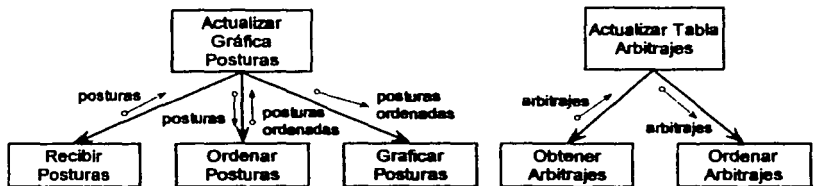
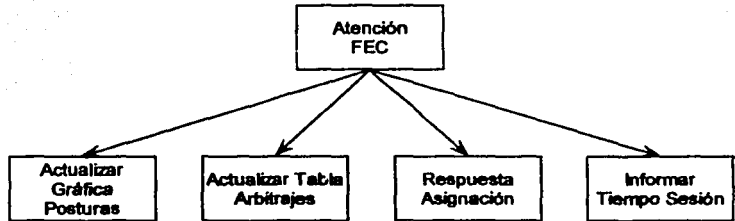
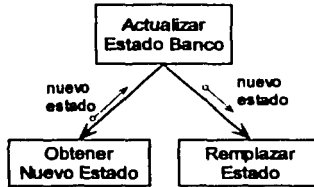
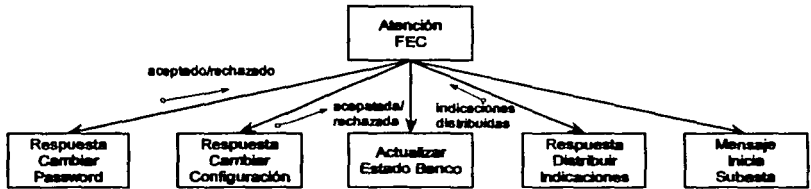


Figura 3.5 Diagrama de estructuras (continuación). División del módulo de Atención al FEC.

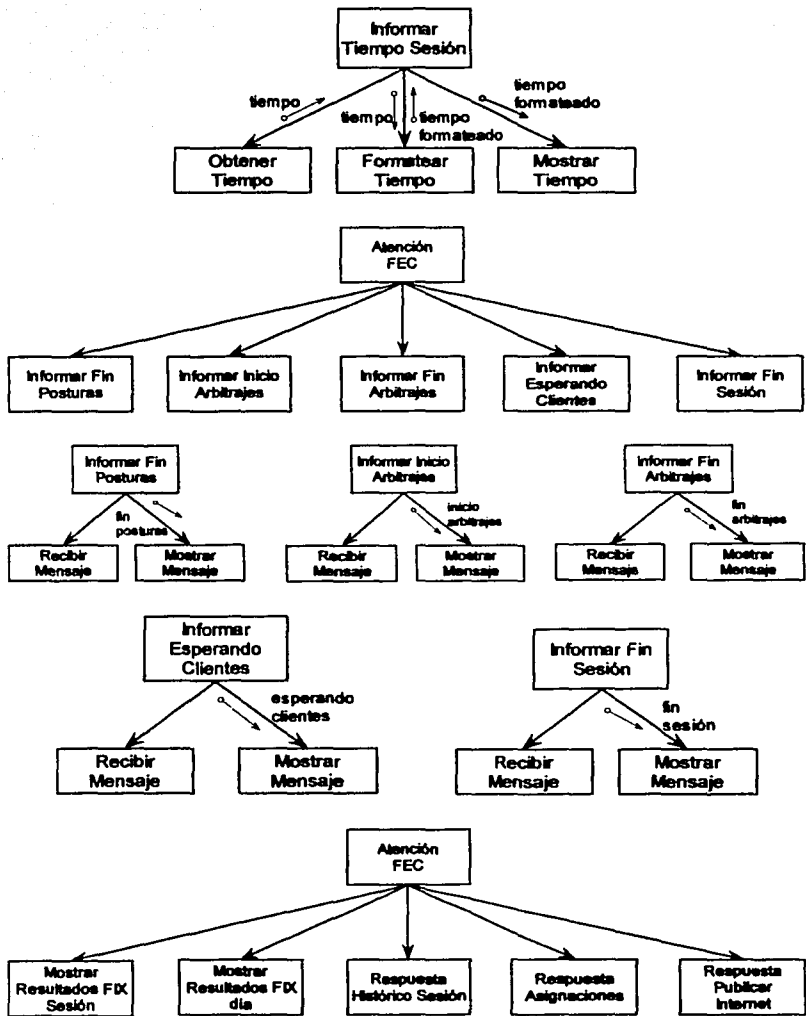


Figura 3.5 Diagrama de estructuras. División del módulo de Atención al FEC (continuación).

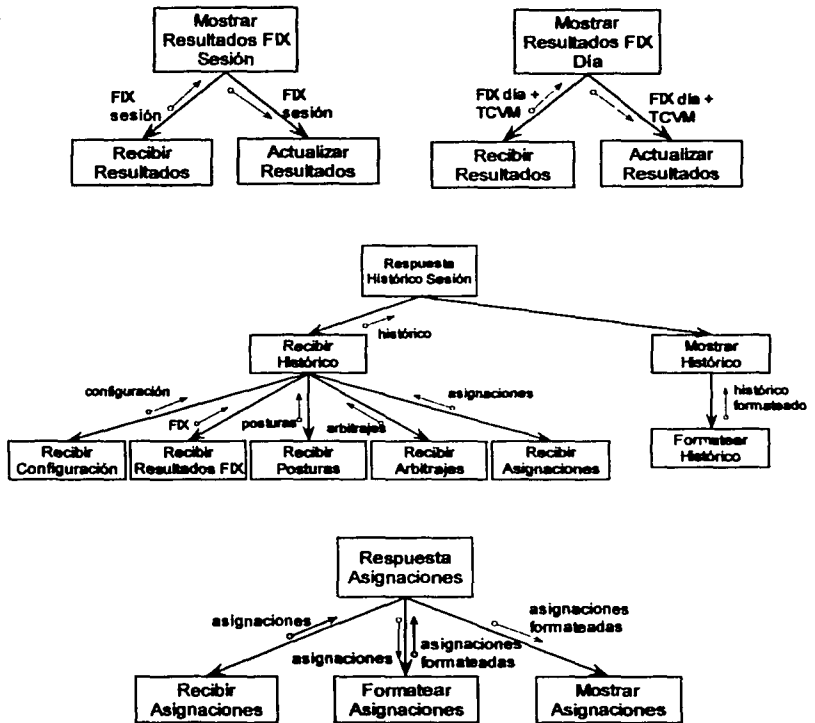


Figura 3.5 Diagrama de estructuras. División del módulo de Atención al FEC (continuación).

IMPLEMENTACIÓN

En este capítulo se describe el desarrollo de las tareas y módulos del sistema FixBan 2.0 cliente Banxico, y su integración en un sistema completo. Se muestran tanto el código realizado en Visual Basic 5.0 como las pantallas resultantes.

Para el manejo de la información de los bancos se crearon tres clases, las cuales se definen al final del capítulo.

4.1 La tarea de comunicaciones

Es llevada a cabo por un grupo de módulos que se encargan de iniciar y mantener la conexión con el FEC a través de sockets. Para establecer la conexión es necesario especificar la dirección IP y el puerto de comunicación (Inicia_Conexión). Se habilita un control de tiempo para permitir que la conexión se intente varias veces antes de dar por fallida la operación (tmrConexion_Timer). Una vez establecida la conexión los datos son recibidos y puestos en estructuras de datos para su manejo e interpretación. En caso de falla en la conexión, se captura el tipo de error (Winsock1_Error) para darlo a conocer y se cierra la conexión (Winsock1_close). Por último, la función de desconexión se encarga de informar al FEC de la salida del operador, se cierra el socket y se guardan los datos de la dirección IP en el registro del sistema. Esta tarea es realizada a través de las siguientes funciones:

```

'*****
' Inicia_Conexion().Intenta la conexión en todos los puertos
' especificando la dirección IP del servidor y el puerto a
' conectarse. En caso de falla, muestra la forma de desconexión
' identificando en num_mens el tipo de problema.
'*****
Public Sub Inicia_Conexion()
    If pto < PtoFin Then
        pto = pto + 1
    Else
        num_mens = 1
        FrmDesconecta.Show
    End If

    Winsock1.Connect var_servidor, pto 'PtoIni
    tmrConexion.Enabled = True
End Sub

'*****
' tmrConexion_Timer. Se activa cuando el lapso de tiempo
' establecido para intentar la conexión ha terminado. Si el
' socket no está conectado, carga la forma de desconexión
'*****

```

```

' indicando en num_mens el tipo de error.
'*****
Private Sub tmrConexion_Timer()
    If Winsock1.State <> sckConnected Then
        num_mens = 1
        FrmDesconecta.Show
    End If
End Sub

'*****
' Asigna a soc la conexión, se guardan los nuevos datos en el
' registro. Envía al FEC la operación de INICIATALK con el
' SERVICIOFIX.
'*****
Private Sub Winsock1_Connect()
    Dim r As Integer

    tmrConexion.Enabled = False
    soc = Winsock1.SocketHandle

    GuardaDatosRegistro

    MsgOut.TipoOp = Chr$(INICIATALK)
    MsgOut.Destiny = Chr$(SERVICIOFIX)
    r = Manda(soc, MsgOut)
End Sub

'*****
' Indica la llegada de datos en el socket. Obtiene los bytes y
' llama a la función AtiendeMensaje.
'*****
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim buffer2 As String
    Dim i As Integer
    Dim bien As Integer
    Dim haybytes, recibio As Integer

    bien = 0
    haybytes = 0
    recibio = 0
    buffer2 = ""
    Winsock1.GetData buffer2

    Do While bien <> 1
        recibio = RecibeVB(buffer2, bytesTotal, MsgIn,
            haybytes, "", 0)
        Select Case recibio
            Case -1
                StatusBar1.Panels(1).Text = "Error en la función
                    RecibeVB"
            Case 0:
                If haybytes = 0 Then
                    Call AtiendeMensaje
                    bien = 1
                Else
                    Call AtiendeMensaje
                End If
        End Select
    End Do

```

TESIS CON
 FALLA DE ORIGEN


```

Case 1
  StatusBar1.Panels(1).Text = "Error al reservar memoria
                             para arreglos internos"
      bien = 1
End Select
Loop
End Sub

```

4.2 La tarea de atención al FEC

Esta tarea comprende todos los módulos que se encargan de procesar los mensajes enviados por el FEC. Dichos módulos son administrados a través de una función llamada `AtiendeMensaje()`, la cual recibe el tipo de operación solicitada por el FEC y ejecuta las funciones necesarias para procesar cada petición.

Las peticiones del FEC son un conjunto de funciones finitas reconocidas por un identificador, sin embargo, si se recibe alguna petición no auténtica ésta es rechazada.

Es importante hacer notar que aunque en el modelo de implementación del programa, el manejo del mensaje "estas vivo" se incluía en la tarea de comunicaciones, para facilitar la programación se decidió incluirlo dentro de la tarea de atención al FEC.

```

'*****
' Atiende los mensajes recibidos en el socket enviados por el FEC'
'*****
Public Sub AtiendeMensaje()
Dim cadena As String
Dim TipoOp As String * 1

    TipoOp = MsgIn.TipoOp

    Select Case TipoOp
    Case Chr$(LOGINREQ)
        var_pregunta_login = var_pregunta_login + 1

        If var_pregunta_login > 1 Then
            pictLogin.Visible = True
            RevisaLogin(PreguntaLogin - 1)
        Else
            EnviaLogin
        End If

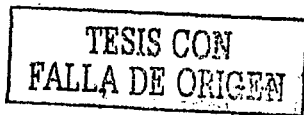
    Case Chr$(PSSWDREQ)
        EnviaPassword

    Case Chr$(LOGINFAIL)
        RevisaLogin (PreguntaLogin)

    Case Chr$(LOGGED)
        LimpiaCajasAlInicio
        HabilitaCajas

    Case Chr$(DENTROSESION)

```



```

EnviaVersion

Case Chr$(VERSIONINCORRECTA)
var_version_incorrecta = True

Case Chr$(DEADSRVR)
DesconectaSocket
num_mens = 2
FrmDesconecta.Show

Case Chr$(SERVFUERADESERVICIO)
num_mens = 3
FrmDesconecta.Show

Case Chr$(AREYOUALIVE)
Beep
MsgOut.TipoOp = Chr$(IMALIVE)
MsgOut.Destiny = Chr$(SERVICIOFIX)

If (Manda(soc, MsgOut) < 0) Then
    StatusBar1.SimpleText = "mal en vivo"

Case Chr$(PSSWDFAIL)
mimsgbox "El password no pudo ser cambiado", , _
t_img_error, "Aviso"

Case Chr$(CONFPASSWD)
var_password = var_nuevo_password
mimsgbox "El password ha sido cambiado", , , _
"Aviso"

Case Chr$(CONFIGURACIONINICIAL)
ActualizaConfEntrada
FrmConvoDatos.ActualizaVariables 1
ActualizaFormaPrincipal

Case Chr$(CONFIGURACIONACEPTADA)
mimsgbox "La nueva configuración fue aceptada.", , , _
"Información"

Case Chr$(CONFIGURACIONRECHAZADA)
mimsgbox "La nueva configuración fue rechazada.", , , _
"Error"

Case Chr$(VALORESANTERIORES)
ActualizaValoresAnteriores

Case Chr$(LISTABANCOS)
ActualizaListaBancos

If (var_sesion_estado = ENTIEMPO) Or
(var_sesion_estado = ENARBITRAJES) Or _
(var_sesion_estado = ESPERANDOCLIENTES) Then
    Bancosfix.CreaBcosPosturas
    Bancosfix.PonBcosSesionActiva
End If

```

```

If (CommDarP.Enabled = True) And
  (CommDarP.Visible = True) Then
  CommDarP.SetFocus
ElseIf (CommQuitarP.Enabled = True) And
  (CommQuitarP.Visible = True) Then
  CommQuitarP.SetFocus
End If

Case Chr$(LLEGABANCO)
  LlegABco

Case Chr$(SEVABANCO)
  SeVaBco

Case Chr$(PRINCIPIASESION)
  barraBotones.Buttons("abrirSesion").Enabled = False
  var_sesion_actual = MsgIn.ArrayD(1)
  var_sesion_estado = ENTIEMPO
  TxtMonto.Text = Str$(MsgIn.ArrayD(2))
  txtSesionActiva.Text = Str$(var_sesion_actual)

  DaParticipacion

  FrmPrincipal.CreaBancosPosturas
  'mmsgbox "Inicia la subasta", , , _
    "¡AVISO!", , , t_sonido_largo, _
    vbHighlightText, vbHighlight
  Dim via As String, comando As String
  via = IIf(Right$(App.Path, 1) = "\", _
    App.Path, App.Path + Trim$("\"))
  comando = via + "EmpiezaSesion.exe" + Space(1)
  comando = comando + "Empieza_la_sesion" + Space(1)
  comando = comando + via + var_chicharra
  Call Shell(comando, vbNormalFocus)

  FrmPosturas.Show

Case Chr$(POSTURARECIBIDA)
  PintaPostura

Case Chr$(TIEMPOESTANTE)
  PintaTiempo (MsgIn.ArrayD(1))

Case Chr$(FINTIEMPOPARTICIPACION)
  PintaTiempo (0)

If Bancosfix.TodasLasPosturas Then
  mmsgbox "El tiempo de participación " + _
    "ha terminado.", , , _
    "AVISO", , , , _
    vbHighlightText, vbHighlight
Else
  mmsgbox "El tiempo de participación ha " + _
    "terminado." + Chr$(10) + _
    "Pero no se han recibido todas " + _
    "las posturas.", , , _
    t_img_error, "Estado de la sesión", _

```

```

        , , vbHighlightText, vbHighlight
    End If

    Case Chr$(COMIENZAARBITRAJE)
        var_sesion_estado = ENARBITRAJES
        PonTiempoArbitraje (10)

    Case Chr$(TIEMPOARBITRAJES)
        Dim tiempo As Integer
        tiempo = MsgIn.ArrayD(1)
        PonTiempoArbitraje (tiempo)
        PintaTiempo (tiempo)

    Case Chr$(ARBITRAJESPOSTURAS)
        PonArbitrajes

    Case Chr$(ARBITRAJEACEPTADO)
        mmsgbobox "El arbitraje ha sido aceptado.", , , _
        "AVISO", , , _
        vbHighlightText, vbHighlight
        PintaOperaciones

    Case Chr$(ARBITRAJERECHAZADO)
        mmsgbobox "El arbitraje ha sido rechazado.", , _
        t_img_error, "IMPORTANTE", , , _
        , vbHighlightText, vbHighlight

    Case Chr$(FINARBITRAJE)
        PonTiempoArbitraje (10)
        PintaTiempo (0)

        If Not Bancosfix.TodasLasPosturas Then
            var_sesion_estado = ESPERANDOCIENTES
        End If

    Case Chr$(POSTURASVIGENTES)
        ActualizaPosturas
        FrmPosturas.Show

    Case Chr$(FINSESION)
        var_sesion_estado = PASIVA
        mmsgbobox "La sesión ha terminado", , , _
        ";AVISO!", , , _
        vbHighlightText, vbHighlight
        Bancosfix.FinalizaSesion
        Bancosfix.ActualizaPermitidos
        BorraBcosPosturas
        LimpiaResultados

    Case Chr$(FIXSESION)
        Dim decimales As Integer, estefix As Long
        decimales = MsgIn.ArrayD(2)
        estefix = MsgIn.ArrayL(1)
        NuevoFix decimales, estefix

    Case Chr$(RECIBEFIXES)
        ActualizaFixes

```

```

Case Chr$(FIXDIARIO)
PonFixDiario

Case Chr$(ASIGNACIONES)
PonAsignaciones

If bandera_impresión Then
FrmAsignaciones.Imprimir
bandera_impresión = False
Else
FrmAsignaciones.Show
End If

Case Chr$(INFOSESION)
FrmHistorias.Show
FrmHistorias.LimpiaHistoria
FrmHistorias.PonInfoSesion

Case Chr$(INTERNETHECHO)
If MsgIn.ArrayD(1) Then
mmsgbox "Se han publicado los hechos " + _
"en Internet", , , _
"Hechos en Internet"
Else
mmsgbox "No se pudieron publicar los " + _
"hechos en Internet", , t_img_error, _
"Error: Hechos en Internet"
End If

Case Chr$(RECIBEMSG)
mmsgbox "Mensaje recibido", , , _
"Información"

Case Chr$(RECIBECINTILLA)
PonCintilla (Trim$(MsgIn.ArrayS(1)))

Case Chr$(INFOSESIONCORTO)
bandera_ultimo = False
GrabaInfoSesionCorto

If bandera_ultimo Then
ImprimeTodasSesiones
mmsgbox "La impresión ha sido terminada"
End If

Case Else
StatusBar1.SimpleText = "no te conozco" +
Str$(Asc(TipoOp))
mmsgbox "Recibo mensaje desconocido : " +
Str$(Asc(TipoOp)), _
t_aceptar, t_img_erFor, "Error no fatal", _
t_no_espera_respuesta

End Select

Refrescar

```

End Sub

4.3 La tarea de atención al operador

Los módulos comprendidos en ésta tarea son ejecutados principalmente a través de controles en forma de menús y botones, como parte de la interfaz gráfica del usuario (GUI). Los módulos comprenden las funciones que el operador requiere para ejecutar las subastas y controlar la operación del cálculo del FIX.

4.3.1 La pantalla de entrada

Esta es la pantalla de inicio y entrada al sistema. El operador debe seleccionar la dirección IP donde se encuentra el FEC, de la caja de selección. El login es BANNICO por default. El operador tiene hasta tres intentos para teclear el password correcto, y un minuto para solicitar la entrada, antes de que la pantalla se cierre. La figura 4.1 muestra la pantalla de entrada.

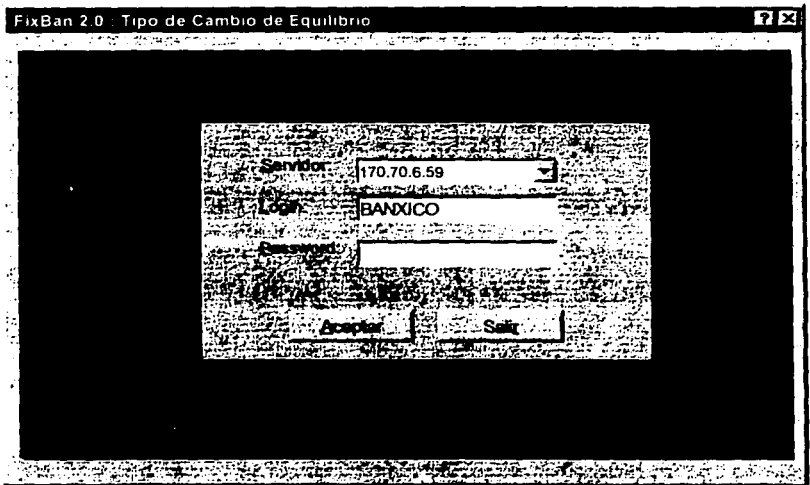


Figura 4.1 La pantalla de entrada.

```

'*****
' commAceptar_Click. Botón Aceptar de la pantalla de entrada.
' Envía los datos obtenidos en login y password para conectarse
' con el FEC.
'*****
Private Sub commAceptar_Click()
Dim i As Integer, esta As Boolean

    On Error GoTo heaven

    If (Trim$(cmbServidor.Text) = "") Or
        (Trim$(txtLogin.Text) = "") Then
        MsgBox "Cajas vacías" + Chr$(10) + _
            "Escriba servidor y login", , _
            t_img_error, "Error"

        Exit Sub
    End If

    pictLogin.Visible = False

    For i = 0 To cmbServidor.ListCount - 1
        If Trim$(cmbServidor.Text) =
            Trim$(cmbServidor.List(i)) Then
            esta = True

            Exit For
        End If
    Next

    If Not esta Then cmbServidor.AddItem cmbServidor.Text, 0

    var_login = txtLogin.Text
    var_password = txtPassword.Text
    var_servidor = Trim$(cmbServidor.Text)

    If var_pregunta_login = 0 Then
        Inicia_Conexion
    Else
        EnviaLogin
    End If

    Exit Sub

heaven:
    MsgBox "Ocurrió la siguiente excepción :" + _
        Str$(Err.Number) + Err.Description + vbCrLf + _
        "Llamar al área de sistemas" + vbCrLf + _
        "Extensiones 8644 y 6y49", vbCritical, "Error crítico"

End Sub

```

4.3.2 La pantalla principal

Esta pantalla muestra los controles principales del sistema. El operador tiene la posibilidad de configurar una sesión, iniciar una subasta, imprimir resultados, crear cartas, solicitar publicación de resultados en Internet, salir del sistema, etc. La figura 4.2 muestra la pantalla principal.

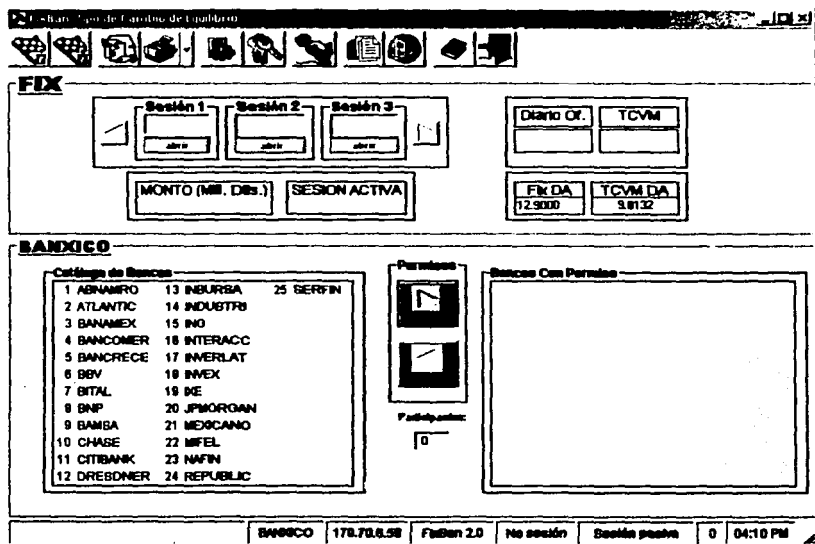


Figura 4.2 La pantalla principal.

Se muestran dos tablas y botones de selección / desección para el manejo de los bancos participantes. Cada banco puede ser seleccionado con el botón derecho del mouse para obtener su clave y teléfono.

Se muestra una barra de estado con la información pertinente del estado de la sesión y un área para despliegue de marquesinas.

Se muestran los últimos tres resultados del FIX obtenido en las sesiones del día, en caso de haber mas de tres sesiones, el operador puede usar los botones con flechas para desplazarse hacia adelante o hacia atrás y solicitar obtener el histórico referente a la sesión correspondiente a dicho resultado del FIX.

```
.....
' Declaración de variables.
.....
VERSION 5.00
Attribute VB_Name = "FrmPrincipal"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Option Explicit
Private Const ID_BANKICO = 9987
Private Bancosfix As cManejaBcos

Private var_sesion_actual As Integer
Private var_sesion_estado As Integer
Private var_sesion_actual_tmp As Integer
Private var_sesion_estado_tmp As Integer

Private Const PtoIni = 7056 'Puerto inicial
Private Const PtoFin = 7060

Private fixes_realizados() As Double
Private decimales_fixes() As Integer
Private NumFixes As Integer
Private FixInicial As Integer

Private var_fix_dia As String
Private var_tcvm_dia As String
Private fix_dia As Long
Private decimales_fix_dia As Integer

Private var_pregunta_login As Integer
Private var_version_incorrecta As Boolean
Private bandera_ultimo As Boolean
Private cadena_cintilla As String
Private apuntador_cintilla As Integer
Private bandera_impresión As Boolean

Private telefonos() As String
Private Const var_chicharra = "EmpiezaSesion.wav"

.....
' Form_Load. Carga la forma principal.
.....
Private Sub Form_Load()
Dim cad As String
Dim i As Integer

    cadena_cintilla = "Este es un mensaje de prueba de la
    cintilla del fixban 2.0"
```

```

var_version_incorrecta = False
var_sesion_estado = -1

'Creo la clase Bancosfix y la inicializo
Set Bancosfix = New cManejaBcos
Set Bancosfix.BotonDarPermiso = CommDarP
Set Bancosfix.BotonQuitarPermiso = CommQuitarP
Set Bancosfix.txtPorParticipar = txtPorParticipar
Set Bancosfix.MallaCatalogo = grCatalogoBcos
Set Bancosfix.MallaPermitidos = grPermisoBcos
Set Bancosfix.MallaPosturas = FrmPosturas.grPosturas

Set Bancosfix.PanelSeleccionados =
    StatusBar1.Panels("nBancosSeleccionados")
Set Bancosfix.PanelComunicaciones =
    StatusBar1.Panels("informacion")

NumFixes = 0
FixInicial = 1
NumBancos = 50

inicializa_formatos
inicializa_pantalla

FrameFix.Width = FrmPrincipal.Width
FrameBcos.Width = FrmPrincipal.Width
'Inicializo el puerto
pto = PtoIni - 1

mnuInformacion.Visible = False
End Sub

```

4.3.3 La pantalla de la configuración

En esta pantalla se pueden cambiar los datos de configuración de las sesiones, tal como: definir el número de sesiones en el día, el monto a subastar, el mínimo número de participantes, la divisa, el tiempo de duración de las posturas y los decimales a usar.

El operador puede decidir entre cambiar la configuración por default, cambiarla para la sesión siguiente o cancelar la operación. La figura 4.3 muestra la pantalla de la configuración.

Configuración

Datos de la Sesión 1

Monto: (millones)

Divisa:

Duración: (segundos)

Mínimo de Participantes:

Número de Dígitos Decimales:

Guardar

Cancelar

Abrir Sesión

Monto subastado

Figura 4.3 La pantalla de la configuración.

```

' Archivo: FrmConvoDatos.frm
' Descripción: Define las funciones para cambiar la
' configuración de las subastas.
'-----
' Declaración de variables
'-----
Option Explicit
Private cambiaron_datos As Boolean
Private guardo_datos As Boolean
Private abre_sesion As Boolean
Private controles() As Control
Private max_sesiones As Integer
Private datos_sesion As Datos_Convocatoria

'-----
' commAbrirSesion_Click. Botón para iniciar una subasta.
'-----
Private Sub commAbrirSesion_Click()
Dim r As Boolean
Dim NPar As Integer
Dim msg, estilo As String, Respuesta As Integer,
cadena As String

estilo = vbYesNo + vbCritical + vbDefaultButton2

```

```

titulo = "IMPORTANTE"

'datos válidos
If Verifica = True Then
    NPar = FrmPrincipal.Nparticipantes

    If NPar < val(TxtParticipantes.Text) Then
        cadena = "El número de bancos con permiso es " _
            + Str$(NPar) _
            + ", el cual es menor que el número de _
            participantes" _ + " de la convocatoria _
            " + Chr$(10) + "¿Desea continuar?"

        Respuesta = mimsgbox(cadena, t_aceptar_cancelar, _
            t_img_pregunta, titulo, t_espera_respuesta)

        If Respuesta = 2 Then
            FrmPrincipal.BorraBcosPosturas
            FrmConvoDatos.Hide
            Exit Sub
        End If

    End If

    MsgOut.Destiny = Chr$(SERVICIOFIX)
    MsgOut.TipoOp = Chr$(INICIARSESION)
    MsgOut.ArrayD(1) = val(TxtMonto.Text)
    MsgOut.ArrayD(2) = val(TxtDuracion.Text)
    MsgOut.ArrayD(3) = val(TxtParticipantes.Text)
    MsgOut.ArrayD(4) = val(TxtDecimales.Text)
    MsgOut.ArrayC(1) = 1

    If (Manda(soc, MsgOut) < 0) Then
        FrmPrincipal.StatusBar1.SimpleText =
            "error envia inicia sesion"
    End If

    End If
    FrmConvoDatos.Hide
End Sub

'*****
' Form Load. Carga la forma FrmConvoDatos.frm e inicializa
' variables.
'*****
Private Sub Form_Load()
    ReDim controles(1 To 5)
    Set controles(1) = TxtMonto
    Set controles(2) = TxtDuracion
    Set controles(3) = TxtParticipantes
    Set controles(4) = TxtDecimales
    Set controles(5) = TxtNSesiones
End Sub

```

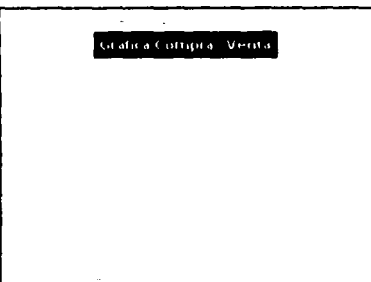
4.3.4 La pantalla de ejecución de las subastas

Esta pantalla muestra el desarrollo de una subasta. La tabla contiene tanto los nombres de los bancos que están participando como sus últimas posturas. Se muestra una gráfica con los valores de las últimas posturas sometidas y una tabla que contiene los arbitrajes formados en caso de existir. Se provee un reloj que le indica al operador el tiempo restante de la sesión.

El operador puede seleccionar los arbitrajes existentes y someterlos para las asignaciones del día. La figura 4.4 muestra la pantalla de ejecución de las subastas.


Sesión pasiva

Gráfica Compra - Venta





COTIZACIONES		
COMPRA	VENTA	BANCO

ARBITRAJES

VENDO A LA COMPRA	COMPRO A LA VENTA	ARBITRAJE MANUAL	
		VENDO A LA COMPRA	COMPRO A LA VENTA
		BANCO	PRECIO
		PRECIO	BANCO

Tiempo Restante





Regresar

OPERACIONES

VENDO A LA COMPRA	COMPRO A LA VENTA

Figura 4.4 La pantalla de las subastas.

```
-----
Archivo:      FrmPosturas.frm
Descripción:  Se encarga de desplegar la subasta activa.
-----
```

```
*****
' Declaración de variables
*****
```

```
Option Explicit
Private Const NUM_SERIES = 2
Private Const SERIE1 = 1
Private Const SERIE2 = 2

Private var_num_ptos As Double
Private var_num_posturas As Integer

Private var_arreglo_compras() As Double
Private var_arreglo_ventas() As Double

Private ArregloPosturas() As Double

Private var_operacion_bco_compra As String
Private var_operacion_compra As String
Private var_operacion_venta As String
Private var_operacion_bco_venta As String
```

```
*****
' PonPosturaEnGrafica. Crea la gráfica de posturas.
*****
```

```
Public Sub PonPosturaEnGrafica()
Dim i As Double
Dim punto As Double
```

Me.CreaArreglosPosturas

```
var_num_ptos = 2 * grCompras.Rows
chartPosturas.ChartGroups(1).Styles(1).Symbol.Size = 0
chartPosturas.ChartGroups(1).Styles(2).Symbol.Size = 0

With chartPosturas.ChartGroups(1).Data
    IsBatched = True
    NumSeries = NUM_SERIES
    NumPoints(1) = var_num_ptos
    NumPoints(2) = var_num_ptos
    Layout = cc2dDataArray
End With

With chartPosturas.ChartGroups(1).SeriesLabels
    Add ("Compra")
    Add ("Venta")
End With

chartPosturas.Legend.IsShowing = True
```

```

With chartPosturas.Legend
.   Anchor = oc2dAnchorSouth
.   Font.Size = 7
.   Orientation = oc2dOrientHorizontal
.   Border = oc2dBorder3DOut
End With

i = 0

For punto = 1 To var_num_ptos

    If punto Mod 2 = 0 Then
        i = i + 1
    End If

    With chartPosturas.ChartGroups(1).Data
        .x(SERIE1, punto) = i
    End With

Next

i = 0
ReDim ArregloPosturas(1 To var_num_ptos, 1 To NUM_SERIES)

For punto = 1 To var_num_ptos
    If punto Mod 2 <> 0 Then
        i = i + 1
    End If

    ArregloPosturas(punto, SERIE1) =
        CDb1(var_arreglo_compras(i))
    ArregloPosturas(punto, SERIE2) =
        CDb1(var_arreglo_ventas(i))

Next

chartPosturas.ChartGroups(1).Data.CopyFromArrayIn
(ArregloPosturas)
chartPosturas.ChartGroups(1).Data.IsBatched = False
End Sub

'*****
' AgregaMontoCompra. Crea una tabla con todos las posturas de
' compra en orden descendente.
'*****
Public Sub AgregaMontoCompra(clave As Integer, compra As Double)
Dim postura
Dim i As Integer

For i = 0 To grCompras.Rows - 1
    If Str$(clave) = grCompras.TextMatrix(i, 1) Then
        grCompras.TextMatrix(i, 0) = Str$(compra)
        FrmPosturas.grCompras.Sort =
            flexSortNumericDescending
        Exit Sub
    End If
Next

postura = Str$(compra) & Chr(9) & Str$(clave)

```

```

grCompras.AddItem postura
FrmPosturas.grCompras.Sort = flexSortNumericDescending
End Sub

```

4.3.5 La pantalla de históricos

En esta pantalla el operador tiene acceso a la información más relevante de alguna sesión en el día. Puede guardar la sesión o imprimirla, así como seleccionar los datos a guardar o imprimir. La figura 4.5 muestra la pantalla de históricos.

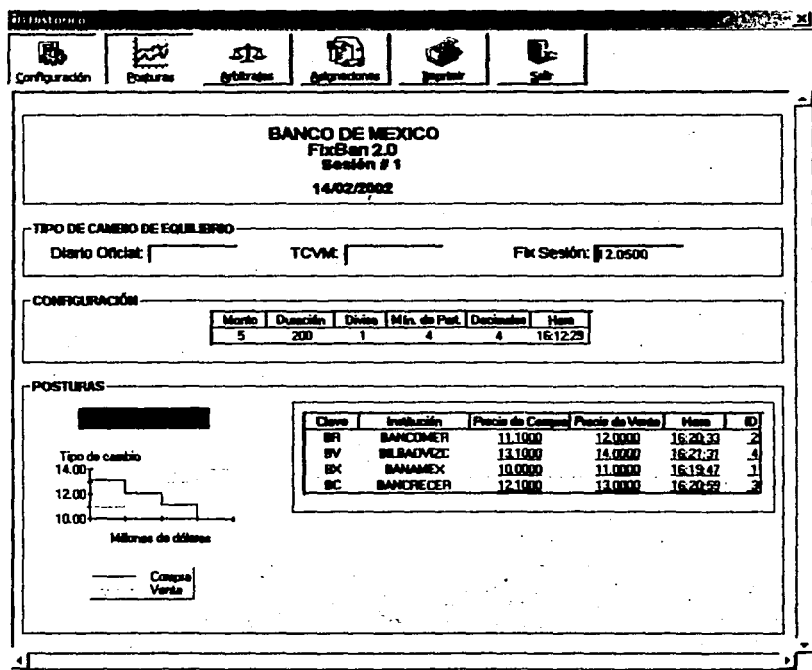


Figura 4.5 La pantalla del histórico.


```

-----
' Archivo:      FrmHistorias.frm
' Descripción: Se encarga de desplegar la historia de
'              una subasta.
-----

```

```

*****
' Declaración de variables.
*****

```

```

Option Explicit
Private Const NUM SERIES = 2
Private Const SERIE1 = 1
Private Const SERIE2 = 2
Private Const DBSPC = (2 * 120)
Private Const MRGNIZQ = 120
Private Const MRGNDR = 120
Private Const MRGNHOR = 120
Private Const MRGNSUP = (2 * 120)
Private Const MRGNINF = (2 * 120)
Private Const MRGNMALLA = 100
Private Const ANCHODEF = 10452
Private Const LARGODEF = 8292

Private var_num_ptos As Double
Private var_num_posturas As Integer
Private var_en_resize As Boolean
Private var_en_cambia_vscroll As Boolean

Private var_alto_original As Long
Private var_ancho_original As Long

```

```

Private ArregloPosturas() As Double '
Private var_sesion As String
Private var_fecha As String
Private var_archivo_datos As String
Private var_bordecito As Long
Private var_primera_vez As Boolean
Private borde_h As Long
Private borde_v As Long
Private var_quitando As Boolean
Private var_frame As Frame

```

```

*****
' PonQuitaConfiguracion. Pone o quita la configuración.
*****

```

```

Private Sub PonQuitaConfiguracion()
  With frConfiguracion
    If .Visible Then
      .Visible = False
    Else
      .Visible = True
    End If

    var_quitando = Not .Visible
  End With

  Set var_frame = frConfiguracion

```

```

        CalculaPosiciones
End Sub

'*****
' Form_Load. Carga la forma FrmHistorias.frm
'*****
Private Sub Form_Load()
    var_primera_vez = True
    var_fecha = Format(Date, "dd/mm/yyyy")
    lbFecha = var_fecha
    var_en_resize = False
    var_en_cambia_Vscroll = False
    var_bordecito = 4 * Screen.TwipsPerPixelX
    Set var_frame = frHeader
End Sub

```

4.3.6 La pantalla de cambio de password

En esta pantalla el operador puede cambiar el password de acceso al sistema. El operador deberá proveer el password anterior para que la operación pueda ser aceptada. El password nuevo debe contener una combinación de mayúsculas y minúscula, y números o caracteres especiales. La figura 4.6 muestra la pantalla de cambio de password.

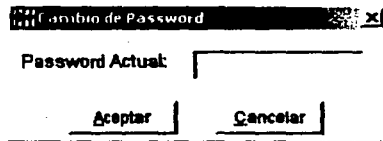


Figura 4.6 La pantalla de cambio de password.

```

-----
' Archivo:      FrmPassword.frm
' Descripción:  Define las funciones para cambiar el password
'-----
'*****
' commAceptar. Botón para iniciar el cambio de password.
'*****
Private Sub commAceptar_Click()
    Select Case var_paso
        Case 1
            'Revisar que sepa el password actual
            If txtTexto.Text = FrmPrincipal.Password Then

```

```

        'es bueno entonces
        var_paso = 2
    Else
        mimsgbox "Password incorrecto.", , _
            t_img_error, "Error de autenticación"
        var_paso = 1
    End If
Case 2
    If Correcto Then
        var_nuevo_password = Me.txtTexto
        var_paso = 3
    Else
        var_paso = 2
    End If
Case 3
    If var_nuevo_password = txtTexto.Text Then
        CambiaPassword
        Unload Me
        Exit Sub
    Else
        var_paso = 2
        mimsgbox "Los passwords no coinciden.", , _
            t_img_error, _
            "Error en el password"
    End If
End Select

txtTexto.SetFocus
End Sub

```

4.3.7 La pantalla de distribución de indicaciones

En esta pantalla el operador puede teclear un mensaje o una marquesina y enviárselo a los bancos. Puede seleccionar entre enviarlo a un banco o a un conjunto de bancos. La figura 4.7 muestra la pantalla de distribución de indicaciones.

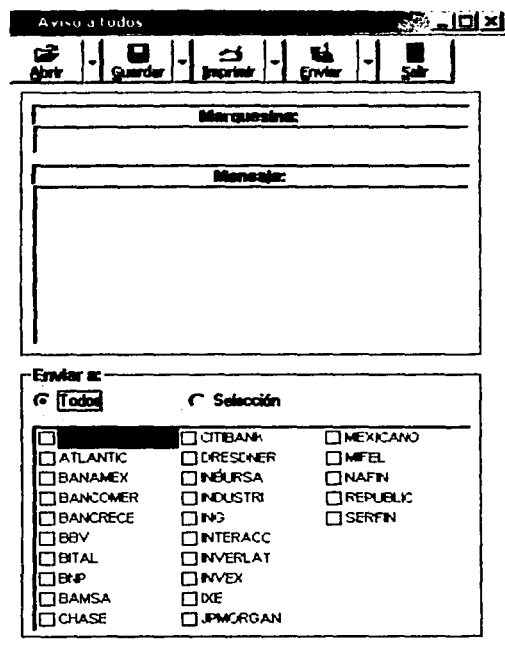


Figura 4.7 La pantalla de distribución de indicaciones.

```

-----
Archivo:      FrmInformacion.frm
Descripción: Define las funciones para distribuir mensajes
              a los bancos.
-----

```

```

*****
' Abrir. Abre un archivo que contiene texto plano.
*****
Private Sub Abrir(cual As aviso)
Dim archivo As String

```

```

CommonDialog1.CancelError = True
On Error GoTo ErrHandler

```

TESIS CON
FALLA DE ORIGEN

```

CommonDialog1.FLAGS = cd1OFNHideReadOnly
CommonDialog1.Filter = "TXT (*.txt)|*.txt"
CommonDialog1.ShowOpen
archivo = CommonDialog1.filename
If cual = t_mensaje Then
    rtfTexto.LoadFile archivo, rtfText
Else
    rtfCintilla.LoadFile archivo, rtfText
End If

Exit Sub

ErrorHandler:
Exit Sub
End Sub

'*****
' Guardar. Guarda en un archivo el texto.
'*****
Private Sub Guardar(cual As aviso)
Dim archivo As String

CommonDialog1.CancelError = True
On Error GoTo ErrorHandler

CommonDialog1.FLAGS = cd1OFNHideReadOnly Or
cd1OFNOverwritePrompt
CommonDialog1.Filter = "TXT (*.txt)|*.txt"
CommonDialog1.ShowSave
archivo = CommonDialog1.filename

If cual = t_cintilla Then
    rtfCintilla.SaveFile archivo, rtfText
Else
    rtfTexto.SaveFile archivo, rtfText
End If

Exit Sub
ErrorHandler:
Exit Sub
End Sub

```

4.3.8 La pantalla de reportes y cartas

Esta pantalla crea la carta que será entregada a Comunicación Social y el comunicado para la subgerencia de Cambios Nacionales. La figura 4.8 muestra la pantalla de generación de reportes.

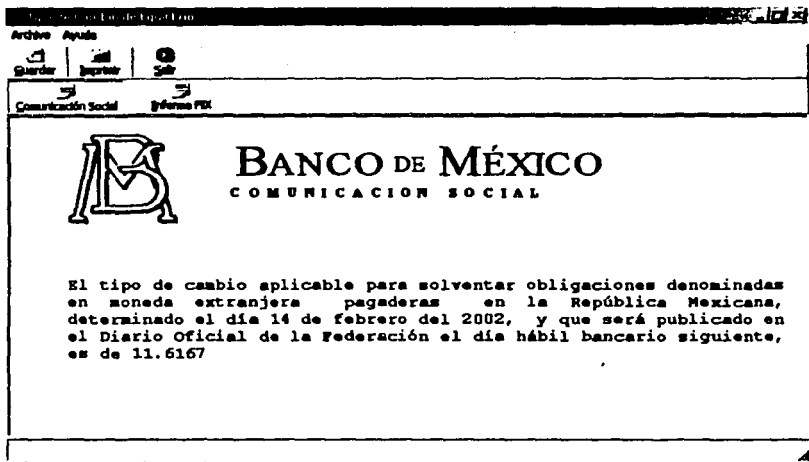


Figura 4.8 La pantalla de generación de cartas. Comunicación Social.

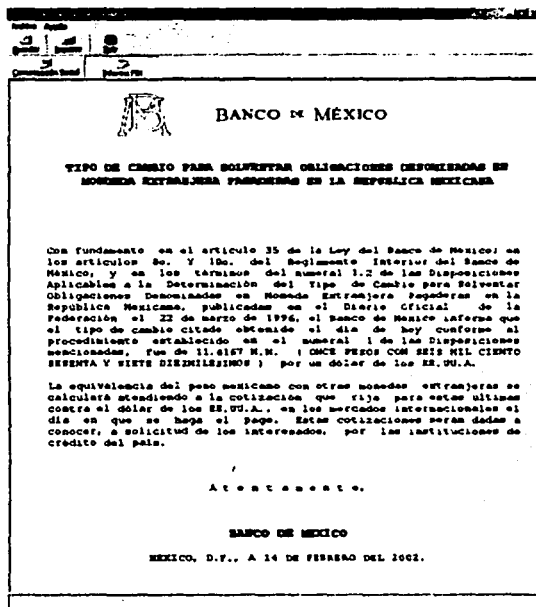


Figura 4.8 La pantalla de generación de cartas (continuación). Comunicado.

```

-----
Archivo:      FrmCarta.frm
Descripción: Define las funciones para generar las cartas.
-----

*****
Declaración de variables
*****
Public var_fix As String

*****
Command1. Botón para salir de la forma.
*****
Private Sub Command1_Click()
    Me.Hide
End Sub

```

```

'*****
' Form_Load. Carga la forma con los datos de un archivo.
'*****
Private Sub Form_Load()
Dim carta As String
    carta = "c:\proyectos\fixban\ComSocialfixBan.rtf"
    frmtexto.rtfTexto.LoadFile carta
End Sub

Private Sub ItemAbrir_Click()
    CommonDialog1.ShowOpen
End Sub

Private Sub ItemGuardarComo_Click()
    CommonDialog1.ShowSave
End Sub

```

4.3.9 La pantalla de impresión de asignaciones

Esta pantalla permite tener acceso a todas las asignaciones ocurridas en el día, guardarlas e imprimirlas. La figura 4.9 muestra la pantalla de impresión de asignaciones.

ASIGNACIONES					
VENDO A LA COMPRA			COMPRO A LA VENTA		
Sesión	Monto	Banco	Precio	Precio	Banco

Figura 4.9 La pantalla de impresión de asignaciones.


```

-----
Archivo:      FrmAsignaciones.frm
Descripción:  Define las funciones para imprimir las
              asignaciones del día.
-----

```

```

*****
' commImprimir. Botón para imprimir.
*****
Private Sub commImprimir_Click()
    Imprimir
End Sub

```

```

*****
' Form_Load. Carga la forma.
*****
Private Sub Form_Load()
Dim cad As String

    cad = ""^Sesión |^Monto |^          Banco
    |< Precio |< Precio |^          Banco
    FrmAsignaciones.GrAsignaciones.Rows = 1
    FrmAsignaciones.GrAsignaciones.FormatString = cad
End Sub

```

```

*****
' ImprimeDatos. Prepara los datos para la impresión.
*****
Public Function ImprimeDatos(sesion As String, cantidad As String,
    PCompra As String, PVenta As String,
    BCompra As String, BVenta As String)

```

```

Dim donde, cad As String
Dim Pos As Integer, tmp As String

```

```

    tmp = LCase$(Mid$(BCompra, 2))
    BCompra = Left$(BCompra, 1) + tmp

    tmp = LCase$(Mid$(BVenta, 2))
    BVenta = Left$(BVenta, 1) + tmp

```

```

    cad = "[fecha]"
    Pos = 1
    Do
        donde = InStr(Pos, rtfOperaciones.Text, cad)
        If donde <> 0 Then
            rtfOperaciones.SelStart = donde - 1
            rtfOperaciones.SelLength = Len(cad)
            rtfOperaciones.SelText =
                LCase$(Format$(Date, "Long Date"))
        End If
    
```

```

        Pos = donde + Len(cad)
    Loop While (donde <> 0)

```

```

    cad = "[sesion]"
    Pos = 1

```

```

Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)
    ' Find string in text.
    If donde <> 0 Then ' If found,
        rtfOperaciones.SelStart = donde - 1
        rtfOperaciones.SelLength = Len(cad)
        rtfOperaciones.SelText = Trim$(sesion)
    End If

    Pos = donde + Len(cad)
Loop While (donde <> 0)

cad = "[cantidad]"
Pos = 1
Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)
    ' Find string in text.
    If donde <> 0 Then ' If found,
        rtfOperaciones.SelStart = donde - 1
        rtfOperaciones.SelLength = Len(cad)
        rtfOperaciones.SelText = Trim$(cantidad)
    End If

    Pos = donde + Len(cad)
Loop While (donde <> 0)

cad = "[PCompra]"
Pos = 1
Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)
    ' Find string in text.
    If donde <> 0 Then ' If found,
        rtfOperaciones.SelStart = donde - 1
        rtfOperaciones.SelLength = Len(cad)
        rtfOperaciones.SelText = Trim$(PCompra)
    End If
    Pos = donde + Len(cad)
Loop While (donde <> 0)

cad = "[BCompra]"
Pos = 1
Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)
    ' Find string in text.
    If donde <> 0 Then ' If found,
        rtfOperaciones.SelStart = donde - 1
        rtfOperaciones.SelLength = Len(cad)
        rtfOperaciones.SelText = Trim$(BCompra)
    End If

    Pos = donde + Len(cad)
Loop While (donde <> 0)

cad = "[PVenta]"
Pos = 1
Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)

```

```

' Find string in text.
If donde <> 0 Then ' If found,
    rtfOperaciones.SelStart = donde - 1
    rtfOperaciones.SelLength = Len(cad)
    rtfOperaciones.SelText = Trim$(PVenta)
End If
Pos = donde + Len(cad)
Loop While (donde <> 0)

cad = "[Bventa]"
Pos = 1
Do
    donde = InStr(Pos, rtfOperaciones.Text, cad)
    ' Find string in text.
    If donde <> 0 Then ' If found,
        rtfOperaciones.SelStart = donde - 1
        rtfOperaciones.SelLength = Len(cad)
        rtfOperaciones.SelText = Trim$(Bventa)
    End If

    Pos = donde + Len(cad)
Loop While (donde <> 0)

rtfOperaciones.SelLength = 0
rtfOperaciones.SelPrint Printer.hDC
End Function

```

4.3.10 La pantalla de selección del mejor arbitraje

Esta pantalla muestra los arbitrajes originados durante la sesión por orden de importancia y le indica al operador cuál es el mejor. La figura 4.10 muestra la pantalla para la selección del mejor arbitraje.

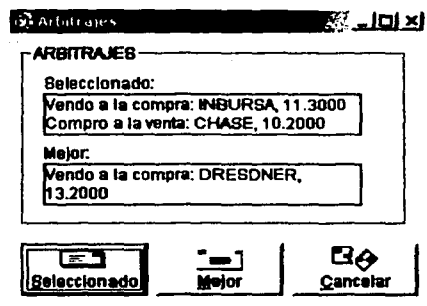


Figura 4.10 La pantalla de selección del mejor arbitraje.

```

-----
Archivo:      FrmArbitrajes.frm
Descripción:  Muestra al operador el mejor arbitraje.
-----

```

Option Explicit

```
Private boton_presionado As String
```

```
Private Sub commCancelar_Click()
    boton_presionado = "cancelar"
    Me.Hide
End Sub
```

```
Private Sub commMejor_Click()
    boton_presionado = "mejor"
    Me.Hide
End Sub
```

```
Private Sub commSeleccionado_Click()
    boton_presionado = "seleccionado"
    Me.Hide
End Sub
```

```
Public Property Get BotonPresionado() As String
    BotonPresionado = boton_presionado
End Property
```

4.3.11 La pantalla de desconexión

```

-----
Archivo:      FrmDesconecta.frm
Descripción:  Muestra un mensaje de desconexión y su causa.
-----

```

TESIS
FALLA DE ORIGEN

```

'-----
'*****
' Command1. Botón para cerrar la forma.
'*****
Private Sub Command1_Click()
    Unload FrmPrincipal
End Sub

'*****
' Form_Load. Carga la forma de desconexión identificando la causa'
'*****
Private Sub Form_Load()
    Timer1.Enabled = True
    Select Case num_mens
    Case 1
        Label1 = "No se pudo conectar.
                Favor de revisar su conexión y que "
                + "el servidor del FEC esté disponible."
    Case 2
        Label1 = "Se cerró la conexión.
                Favor de salir y volver a intentarlo."
    Case 3
        Label1 = "Servicio FixBan no disponible."
    Case 4
        Label1 = "Versión incorrecta del Fix"
                + vbCrLf + "
                Favor de comunicarse al área de Sistemas"
                + vbCrLf + "Extensiones 6749 y 8644"
    End Select
End Sub

'*****
' Timer1. Cierra la aplicación, evitando que se quede activa
' cuando se perdió la conexión.
'*****
Private Sub Timer1_Timer()
    Unload FrmPrincipal
End Sub

```

4.3.12 La pantalla de mensajes

Esta pantalla contiene mensajes pertinentes del sistema como el estado de la sesión de la subasta para el operador. La figura 4.11 muestra la pantalla de información de mensajes.

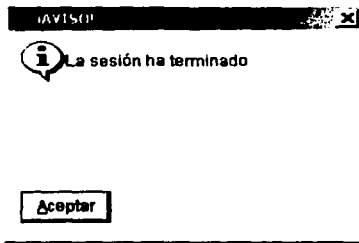


Figura 4.11 La pantalla de información de mensajes.

```

-----
' Archivo: FrmMensajes.frm
' Descripción: Muestra mensajes como el inicio de la subasta.
-----
'*****
' Declaración de variables.
'*****
Option Explicit

'Private Const chicharra = "empiezasesion.wav"
Private Const HWND_TOP = 0
Private Const HWND_BOTTOM = 1
Private Const HWND_TOPMOST = -1
Private Const HWND_NOTOPMOST = -2

Private Const SWP_NO_ACTIVATE = &H10
Private Const SWP_SHOWWINDOW = &H40
Private Const SWP_NOSIZE = &H1
Private Const SWP_NOMOVE = &H2

Private Const FLAGS = SWP_NOSIZE Or SWP_NOMOVE

Private var respuesta As Integer
Private var tipo As Integer
Private var sonido As t_sonidos

'*****
' commBotones. Descarga la forma.
'*****
Private Sub commBotones_Click(Index As Integer)
    var_respuesta = Index
    Unload Me
End Sub

```

TESIS CON
FALLA DE ORIGEN

4.4 Módulos de clase

Se definieron tres clases para el manejo de la información de los bancos. La clase cUnbanco contiene las características de un banco pertinentes al sistema, tales como su nombre, clave, estado y las posturas que ha puesto en el día. Esta clase define también propiedades de los bancos tales como: color, imágenes y posición, utilizadas en la GUI.

La clase cBancos, se diseñó para crear un conjunto de bancos y definir métodos para agregar, quitar, buscar y cambiar el estado de cada banco. Esta clase define una tabla donde se desplegarán las propiedades gráficas de los bancos como nombre, color e imagen.

La clase cManejaBcos, fue creada con el propósito de crear tres conjuntos de bancos, y manejar sus propiedades dependiendo del conjunto al que el banco pertenezca. Estos conjuntos son: el catálogo de bancos, los bancos seleccionados para participar y los bancos que están actualmente participando en la sesión.

```
-----
: Archivo:      banco.cls
: Descripción:  Presenta la clase cUnBanco. Define los
:               atributos de un banco tales como: nombre
:               y clave, estado, posturas, etc. Define
:               los métodos empleados para manejar las
:               propiedades del banco, tales como:
:               obtener y cambiar su estado.
:-----
```

```
*****
'Declaración de atributos.
*****
```

Option Explicit

```
Private Const MASKCONECTADO = 1
Private Const MASKPERMISO = 2
Private Const MASKPARTICIPO = 4
Private Const MASKSESION = 24
Private Const MASKCON1VEZ = 32
Private Const MASKYACOTIZO = 64

Private Const COLORCONECTADO = &H8000&
Private Const COLORDESCONECTADO = &HC0&
Private Const COLORDEFAULT = &H80000005
Private Const LETRAESTILOCONECTADO = 0 'flexTextRaised
Private Const LETRAESTILODESCONECTADO = 0 'flexTextInset
Private Const LETRAESTILODEFAULT = 0 'felxtextflat
Private Const IMAGENSELECCIONADO = 2
Private Const IMAGENYAPARTICIPO = 4
Private Const IMAGENNOCOTIZO = 1
Private Const IMAGENYACOTIZO = 3

Private var_nombre As String
Private var_clave As Integer
Private var_estado As Byte
Private var_sesion As Integer
```

```

Private var_monto_compra As String
Private var_monto_venta As String
Private var_decimales As Integer
Private var_nombre_largo As String
Private var_llave As String
Private var_sesion_estado As Boolean

Private var_muestra_montocompra As Boolean
Private var_muestra_montoventa As Boolean
Private var_muestra_img As Boolean
Private col_compra As Long
Private col_venta As Long
Private col_img As Integer

'Datos de visualizacion
Private var_renglon As Integer
Private var_columna As Integer 'aquí se pone el nombre
Private var_malla As MSFlexGrid

'*****
' EstaConectado. Devuelve verdadero si el banco está conectado,
' sino devuelve falso.
'*****
Public Function EstaConectado() As Boolean
    If (var_estado And MASKCONECTADO) = 1 Then
        EstaConectado = True
    Else
        EstaConectado = False
    End If
End Function

'*****
' Conectar. Cambia el estado del banco a conectado.
'*****
Public Sub Conectar()
    var_estado = (var_estado Or MASKCONECTADO)
    var_estado = (var_estado Or MASKCONIVEZ)
    Actualiza
End Sub

'*****
' DarPermiso. Marca en el estado que tiene permiso.
'*****
Public Sub DarPermiso()
    var_estado = (var_estado Or MASKPERMISO)
    Actualiza
End Sub

'*****
' TienePermiso. Devuelve verdadero si tiene permiso para
' participar, sino falso.
'*****
Public Function TienePermiso() As Boolean
    If (var_estado And MASKPERMISO) = 2 Then
        TienePermiso = True
    Else
        TienePermiso = False
    End If
End Function

```



```
End If
End Function
```

```
*****
' YaParticipo. Devuelve verdadero si ya participó, sino falso.
' *****
Public Function YaParticipo() As Boolean
    If (var_estado And MASKPARTICIPO) = 4 Then
        YaParticipo = True
    Else
        YaParticipo = False
    End If
End Function
```

```
*****
' Participa. Pone en el estado que ya participó.
' *****
Public Sub Participa()
    var_estado = var_estado Or MASKPARTICIPO
    Actualiza
End Sub
```

```
*****
' Cotiza. Pone en el estado que ya cotizó.
' *****
Public Sub Cotiza()
    var_estado = var_estado Or MASKYACOTIZO
    Actualiza
End Sub
```

```
-----
' Archivo:      cbancos.cls
' Descripción:  Contiene la clase cBancos, la cual es
'              un conjunto de bancos y describe métodos,
'              como buscar, agregar y quitar a un banco
'              del conjunto.
' -----
```

```
*****
' Declaración de atributos.
' *****
Option Explicit
```

```
Private Const MAXREN = 12

Private Const ESTILO_MALLA_CATALOGO = 0
Private Const ESTILO_MALLA_PERMITIDOS = 1
Private Const ESTILO_MALLA_POSTURAS = 2

Private bancos As Collection
Private var_malla As MSFlexGrid
Private var_malla_estilo As Integer
```

```
*****
' Class. Inicializa y termina la clase.
' *****
Private Sub Class_Initialize()
```

```

        Set bancos = New Collection
        Set var_malla = Nothing
    End Sub

Private Sub Class_Terminate()
    Set bancos = Nothing
    Set var_malla = Nothing
End Sub

'*****
' Agregar. Agrega un nuevo banco a la colección.
'*****
Public Sub Agregar(cve As Integer, nom As String, nom_largo As
String, llave As String, edo As Byte, sesion As Integer)
Dim banco As cUnBanco

    On Error GoTo NoEsta
        Set banco = bancos(nom)
        banco.nombre_largo = nom_largo
        banco.llave = llave
        banco.clave = cve
        banco.estado = edo
        banco.sesion = sesion
    Exit Sub

NoEsta:
    Set banco = New cUnBanco
    banco.nombre = nom
    banco.nombre_largo = nom_largo
    banco.llave = llave
    banco.clave = cve
    banco.estado = edo
    banco.sesion = sesion
    Select Case var_malla_estilo
    Case ESTILO_MALLA_CATALOGO 'Estilo con número
        banco.MuestraImg = False
    Case ESTILO_MALLA_PERMITIDOS 'Estilo con imagen
        banco.MuestraImg = True
        banco.ColImagen = -1
    Case ESTILO_MALLA_POSTURAS 'Estilo COMPRA|VENTA|BANCO
        banco.MuestraCompra = True
        banco.MuestraVenta = True
        banco.ColCompra = -2
        banco.ColVenta = -1
    End Select

    Dim bcoaux As cUnBanco

    For Each bcoaux In bancos
        If bcoaux.nombre > nom Then
            bancos.Add banco, nom, bcoaux.nombre
            Set banco.malla = var_malla
            Set banco = Nothing
            Actualiza

```

```

Exit Sub
End If
Next

bancos.Add banco, nom
Set banco.malla = var_malla
Set banco = Nothing
Actualiza
End Sub

```

```

-----
Archivo:      cManejaBcos.cls
Descripción:  Esta clase se encarga de crear tres conjuntos
              de bancos. Bancos_catalogo, es aquel que
              tiene todos los bancos inscritos para partici-
              par en el cálculo del FIX. Bancos permitidos
              son los escogidos para participar en una se-
              sión. Bancos_posturas maneja los bancos que
              están participando actualmente.
-----

```

```

*****
'Declaración de atributos.
*****

```

```
Option Explicit
```

```
Private Const ESTILO_MALLA_CATALOGO = 0
Private Const ESTILO_MALLA_PERMITIDOS = 1
Private Const ESTILO_MALLA_POSTURAS = 2
```

```
Private bancos_catalogo As cBancos
Private bancos_permitidos As cBancos
Private bancos_posturas As cBancos
```

```
Private boton_dar As CommandButton
Private boton_quitar As CommandButton
```

```
Private panel_seleccionados As Panel
Private panel_comunicaciones As Panel
```

```
Private txt_por_participar As TextBox
Private malla1 As MSFlexGrid
Private malla2 As MSFlexGrid
```

```

*****
' PonBanco. Pone un banco en la malla del catálogo de bancos o
' en la de bancos para participar, según su estado.
*****

```

```
Public Sub PonBanco(cve As Integer, nom As String, nom_largo
As String, llave As String, edo As Byte, sesion As Integer)
Dim banco As cUnBanco
```

```
Set banco = New cUnBanco
banco.estado = edo
```

```
If banco.TienePermiso = False Then
banco.catalogo.Agregar cve, nom, nom_largo, llave,
```

```

edo, sesion
Else
    bancos_permitidos.Agregar cve, nom, nom_largo, llave,
edo, sesion
End If

Set banco = Nothing
End Sub

```

```

'*****
' ConectaBco. Busca un banco en ambas mallas y lo muestra como
' conectado.
'*****

```

```

Public Sub ConectaBco(cve As Integer)
Dim banco As cUnBanco

```

```

Set banco = bancos_catalogo.Busca(cve)

```

```

If TypeName(banco) <> "Nothing" Then
    banco.Conectar
Set banco = Nothing
Exit Sub
End If

```

```

Set banco = bancos_permitidos.Busca(cve)

```

```

If TypeName(banco) <> "Nothing" Then
    banco.Conectar
Set banco = Nothing
End If

```

```

Set banco = bancos_posturas.Busca(cve)

```

```

If TypeName(banco) <> "Nothing" Then
    banco.Conectar
Set banco = Nothing
End If

```

```

End Sub

```

```

-----
' Archivo:      globals.bas
' Descripción:  Contiene la definición de variables globales
'              y funciones que son usadas por diversos módu-
'              los.
'              Inicializa el registro en el sistema con los
'              datos de dir IP del FEC y carga la pantalla
'              de inicio.
-----

```

```

'*****
' Declaración de funciones externas.
'*****
Attribute VB_Name = "globals"
Declare Function RecibeVB Lib "msgfront.dll"
    (ByVal buffer1 As String, ByVal nbinbu"1 As Integer,
    buf As MsgFront1, haybytes As Byte, ByVal certi As String,

```

```

sizeCrt As Long) As Integer
Declare Function Manda Lib "msgfront.dll"
    (ByVal S As Integer,
     buf As MsgFront1) As Integer
Declare Function Parseo Lib "msgfront.dll"
    (ByVal num_formatos As Integer, arr_formatos As Arreglo,
     tipo_frm As ArrOpFrmt, tipo_firma As ArrOpFirma) As Integer
Declare Function Inicia_Parametros Lib "msgfront.dll"
    (ByVal SizeCads As Integer, ByVal SizeArray As Integer)
    As Integer

```

'Operaciones enviadas por el cliente Banxico

```

Global Const LOGOUT = 0
Global Const LOGIN = 1
Global Const PSSWD = 2
Global Const IMALIVE = 7
Global Const INICIATALK = 16
Global Const DENTROSESION = 13
Global Const MANDAVERSION = 31
Global Const DAPERMISO = 32
Global Const QUITAPERMISO = 33
Global Const INICIARSESION = 34
Global Const FIDELISTABANCOS = 36
Global Const NUEVACONFIGURACION = 37
Global Const PIDEFIXIDIARIO = 38
Global Const SELECCIONAARBITRAJE = 39
Global Const FIDEASIGNACIONES = 40
Global Const FIDEINFOSESION = 41

Global Const MANDAMSG = 43
Global Const MANDAMSGCINTILLA = 44
Global Const PUBLICAHECHOSINTERNET = 45
Global Const MSGALGUNOS = 46 'N(%D)N(%S) CVES, RENGLONES
Global Const CINTILLAALGUNOS = 47 'N(%D)%S
Global Const FIDEINFOCORTO = 48
'' Para enviar el password de banxico para operación
Global Const MANDAPASSWORD = 49

```

```

Global Const CHANGEPPSSWD = 212
Global Const LOGINREQ = 254
Global Const PSSWDREQ = 255

```

'Operaciones recibidas por el cliente Banxico

```

Global Const FIDEPASSWORD = 125
Global Const VERSIONINCORRECTA = 126
Global Const VERSIONCORRECTA = 127
Global Const LISTABANCOS = 128
Global Const PRINCIPIASESION = 129
Global Const TIEMPOESTANTE = 130
Global Const TIEMPOARBITRAJES = 131
Global Const POSTURARECIBIDA = 132
Global Const FINSESION = 134
Global Const RESULTADO = 135
Global Const LLEGABANCO = 138
Global Const SEVABANCO = 139
Global Const CONFIGURACIONINICIAL = 140

```

```

Global Const CONFIGURACIONACEPTADA = 141
Global Const CONFIGURACIONRECHAZADA = 142
Global Const COMIENZAARBITRAJE = 143
Global Const FINARBITRAJE = 144
Global Const FINTIEMPOPARTICIPACION = 145
Global Const FIXSESION = 146
Global Const FIXDIARIO = 147
Global Const ARBITRAJESPOSTURAS = 148
Global Const RECIBEFIXES = 150
Global Const POSTURASVIGENTES = 151

```

```

Global Const ARBITRAJEACEPTADO = 153
Global Const ARBITRAJERECHAZADO = 154
Global Const ASIGNACIONES = 156
Global Const INFOSESION = 157
Global Const VALORESANTERIORES = 159
Global Const RECIBEMSG = 160
Global Const RECIBECINTILLA = 161
Global Const INTERNETHECHO = 162
Global Const RECIBEMSGALGUNOS = 163 'N(%S)
Global Const RECIBECINTILLAALGUNOS = 164 '(%S)
Global Const INFOSESIONCORTO = 165
Global Const CONFPASSWD = 236
Global Const DEADSRVR = 243
Global Const AREYOUALIVE = 245
Global Const PSSWDFAIL = 249
Global Const LOGINFAIL = 251
Global Const SERVFUERADESERVICIO = 252
Global Const LOGGED = 253

```

```

'-----
' Se define una estructura para el manejo de los mensajes del
' protocolo de comunicación con el FEC.
'-----

```

```

Type MsgFront1
  Destiny As String * 1           'destinatario del mensaje
  TipoOp As String * 1           'tipo de operación
  ArrayD(1 To ArraySize) As Integer 'arreglo de enteros
  ArrayC(1 To ArraySize) As Byte   'arreglo de caracteres
  ArrayL(1 To ArraySize) As Long    'arreglo de longs
  ArrayN(1 To ArraySize) As Long    'arreglo de índices
  ArrayS(1 To ArraySize) As String * stringSize 'arreglo de cadenas
End Type

```

```

'*****
' inicializa_formatos. Se encarga de inicilizar los valores de
' los formatos usados para el intercambio de mensajes con el FEC.
'*****
Public Sub inicializa_formatos()
  Dim Format As Arreglo
  im OpFr As ArrOpFrmt
  im OpFma As ArrOpFirma
  'Los formatos se numeran a partir del 1

```

```

Format.frmt (1) = "ccd" & Chr$(0)
Format.frmt (2) = "s" & Chr$(0)
Format.frmt (3) = "d" & Chr$(0)
Format.frmt (4) = "ss" & Chr$(0)
Format.frmt (5) = "ddcdd" & Chr$(0)
Format.frmt (6) = "dddcddd" & Chr$(0)
Format.frmt (7) = "ddl" & Chr$(0)
Format.frmt (8) = "n(dcsssd)" & Chr$(0) 'n
Format.frmt (9) = "ddl" & Chr$(0)
Format.frmt (10) = "c" & Chr$(0) 'no usado
Format.frmt (11) = "ddn(dldl)" & Chr$(0)
Format.frmt (12) = "n(dl)" & Chr$(0)
Format.frmt (13) = "dn(dll)" & Chr$(0)
Format.frmt (14) = "dddcddd" & Chr$(0)
Format.frmt (15) = "dldl" & Chr$(0)
Format.frmt (16) = "n(ddddldl)" & Chr$(0)
Format.frmt (17) = "ddddcdlldn(dlld)n(dldl)n(dldl)" & Chr$(0)
Format.frmt (18) = "n(s)" & Chr$(0)
Format.frmt (19) = "n(d)" & Chr$(0)
Format.frmt (20) = "dd" & Chr$(0)
Format.frmt (21) = "dldn(dll)" & Chr$(0)
Format.frmt (22) = "n(d)n(s)" & Chr$(0)
Format.frmt (23) = "n(d)s" & Chr$(0)

```

4.1 Pruebas

Se llevaron a cabo diversas pruebas para evaluar el desempeño del sistema, y para comprobar la exactitud de los resultados. A continuación se presentan los resultados de una prueba mostrándose la transición de las subastas siguiendo el proceso descrito para obtener el resultado del FIX.

La Figura 4.12 muestra la pantalla principal donde se muestran los bancos que están inscritos para participar en el cálculo del FIX.

El menú de controles muestra activos aquellos que son válidos durante este estado del sistema, tales como: abrir una sesión (icono con la bandera negra), mostrar el módulo de la subasta (icono con la bandera verde), cargar módulo de impresión (icono con impresora), cambio de configuración (icono con tuercas), envío de mensajes a otros bancos (icono con teléfono), módulo de ayuda (icono con manual), salir del sistema. Se muestran los valores del FIX ocurridos el día anterior (Fix Da y TCVM DA).

Se han seleccionado cuatro bancos (Banamex, Bancomer, Bancrecer y BBV) para la ejecución de la primer subasta.

Windows 95 - Entorno de Trabajo

FIX

<table border="1"> <tr> <td> <table border="1"> <tr> <td>Sección 1</td> <td>Sección 2</td> <td>Sección 3</td> </tr> <tr> <td>abrir</td> <td>abrir</td> <td>abrir</td> </tr> </table> </td> <td> <table border="1"> <tr> <td>Diario Of.</td> <td>TCVM</td> </tr> <tr> <td></td> <td></td> </tr> </table> </td> </tr> <tr> <td> <table border="1"> <tr> <td>MONTO (Mil. Dts.)</td> <td>SESION ACTIVA</td> </tr> <tr> <td></td> <td></td> </tr> </table> </td> <td> <table border="1"> <tr> <td>Fix DA</td> <td>TCVM DA</td> </tr> <tr> <td>12.9000</td> <td>9.8132</td> </tr> </table> </td> </tr> </table>	<table border="1"> <tr> <td>Sección 1</td> <td>Sección 2</td> <td>Sección 3</td> </tr> <tr> <td>abrir</td> <td>abrir</td> <td>abrir</td> </tr> </table>	Sección 1	Sección 2	Sección 3	abrir	abrir	abrir	<table border="1"> <tr> <td>Diario Of.</td> <td>TCVM</td> </tr> <tr> <td></td> <td></td> </tr> </table>	Diario Of.	TCVM			<table border="1"> <tr> <td>MONTO (Mil. Dts.)</td> <td>SESION ACTIVA</td> </tr> <tr> <td></td> <td></td> </tr> </table>	MONTO (Mil. Dts.)	SESION ACTIVA			<table border="1"> <tr> <td>Fix DA</td> <td>TCVM DA</td> </tr> <tr> <td>12.9000</td> <td>9.8132</td> </tr> </table>	Fix DA	TCVM DA	12.9000	9.8132
<table border="1"> <tr> <td>Sección 1</td> <td>Sección 2</td> <td>Sección 3</td> </tr> <tr> <td>abrir</td> <td>abrir</td> <td>abrir</td> </tr> </table>	Sección 1	Sección 2	Sección 3	abrir	abrir	abrir	<table border="1"> <tr> <td>Diario Of.</td> <td>TCVM</td> </tr> <tr> <td></td> <td></td> </tr> </table>	Diario Of.	TCVM													
Sección 1	Sección 2	Sección 3																				
abrir	abrir	abrir																				
Diario Of.	TCVM																					
<table border="1"> <tr> <td>MONTO (Mil. Dts.)</td> <td>SESION ACTIVA</td> </tr> <tr> <td></td> <td></td> </tr> </table>	MONTO (Mil. Dts.)	SESION ACTIVA			<table border="1"> <tr> <td>Fix DA</td> <td>TCVM DA</td> </tr> <tr> <td>12.9000</td> <td>9.8132</td> </tr> </table>	Fix DA	TCVM DA	12.9000	9.8132													
MONTO (Mil. Dts.)	SESION ACTIVA																					
Fix DA	TCVM DA																					
12.9000	9.8132																					

BANQUEO

Catálogo de Bancos <table border="1"> <tr><td>1 ABNAMBRO</td><td>13 INVERLAT</td></tr> <tr><td>2 ATLANTIC</td><td>14 INVEX</td></tr> <tr><td>3 BITAL</td><td>15 DE</td></tr> <tr><td>4 BNP</td><td>16 JPMORGAN</td></tr> <tr><td>5 BANGA</td><td>17 MEXICANO</td></tr> <tr><td>6 CHABE</td><td>18 MFEL</td></tr> <tr><td>7 CITIBANK</td><td>19 NAFIN</td></tr> <tr><td>8 DREBDNER</td><td>20 REPUBLIC</td></tr> <tr><td>9 INBURGA</td><td>21 SERFIN</td></tr> <tr><td>10 INDUSTRI</td><td></td></tr> <tr><td>11 INO</td><td></td></tr> <tr><td>12 INTERACC</td><td></td></tr> </table>	1 ABNAMBRO	13 INVERLAT	2 ATLANTIC	14 INVEX	3 BITAL	15 DE	4 BNP	16 JPMORGAN	5 BANGA	17 MEXICANO	6 CHABE	18 MFEL	7 CITIBANK	19 NAFIN	8 DREBDNER	20 REPUBLIC	9 INBURGA	21 SERFIN	10 INDUSTRI		11 INO		12 INTERACC		Participación <table border="1"> <tr><td><input type="checkbox"/></td></tr> <tr><td><input type="checkbox"/></td></tr> </table> <p>Participación: 4</p>	<input type="checkbox"/>	<input type="checkbox"/>	Bancos Con Partidas <table border="1"> <tr><td>T BANAMEX</td></tr> <tr><td>T BANCOMER</td></tr> <tr><td>T BANCRECE</td></tr> <tr><td>T BEV</td></tr> </table>	T BANAMEX	T BANCOMER	T BANCRECE	T BEV
1 ABNAMBRO	13 INVERLAT																															
2 ATLANTIC	14 INVEX																															
3 BITAL	15 DE																															
4 BNP	16 JPMORGAN																															
5 BANGA	17 MEXICANO																															
6 CHABE	18 MFEL																															
7 CITIBANK	19 NAFIN																															
8 DREBDNER	20 REPUBLIC																															
9 INBURGA	21 SERFIN																															
10 INDUSTRI																																
11 INO																																
12 INTERACC																																
<input type="checkbox"/>																																
<input type="checkbox"/>																																
T BANAMEX																																
T BANCOMER																																
T BANCRECE																																
T BEV																																

BANQUEO 178.70.8.58 FileNet 2.0 No sesión Sesión pasiva 4 04:12 PM

Figura 4.12 Selección de participantes.

Después se configuró la sesión con los siguientes datos: monto = 5, duración = 200, mínimo de participantes = 4 y número de dígitos decimales = 4. La figura 4.13 muestra la pantalla donde se realiza este proceso.

TESIS CON FALLA DE ORIGEN

Configuración

Datos de la Sesión 1

Monto: (millones)

Divisa:

Duración: (segundos)

Mínimo de Participantes:

Número de Dígitos Decimales:

Mensaje de estado

Figura 4.13 Configuración de la subasta.

Una vez que el operador empieza la subasta una pantalla de avisos aparecerá dando a conocer que la subasta ha empezado. La figura 4.14 muestra la pantalla de las posturas donde se muestran los bancos seleccionados. El color verde del banco Banamex indica que está conectado, mientras la ausencia de color del resto significa que no se han conectado durante el día.

La gráfica de compra y venta muestra la postura sometida por Banamex, los datos de la postura se muestran en la tabla de cotizaciones.

TESIS CON
FALLA DE ORIGEN

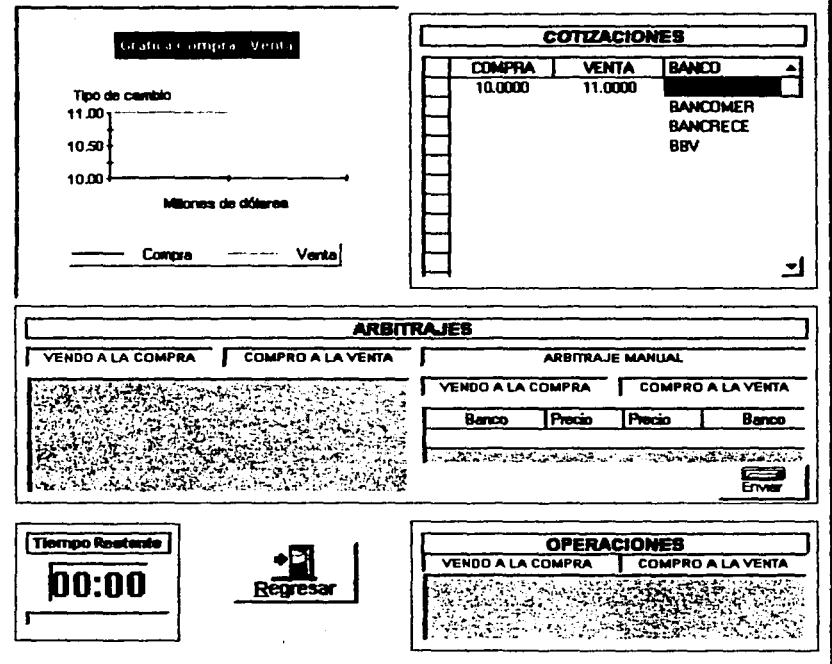


Figura 4.14 Ejecución de la subasta.

La figura 4.15 muestra las posturas de los bancos Bancomer, Banamer y Bancrecer. El color rojo indica que ya se desconectaron, sin embargo, las posturas sometidas al final se quedan para ser usadas en el cálculo del FIX.

El banco en verde BBV aunque está conectado no ha enviado ninguna postura, el tiempo indica 0:00, es decir, que el tiempo para someter las posturas ha terminado y como aún falta una postura para proceder al cálculo del FIX, el sistema pasará al estado de espera de clientes.

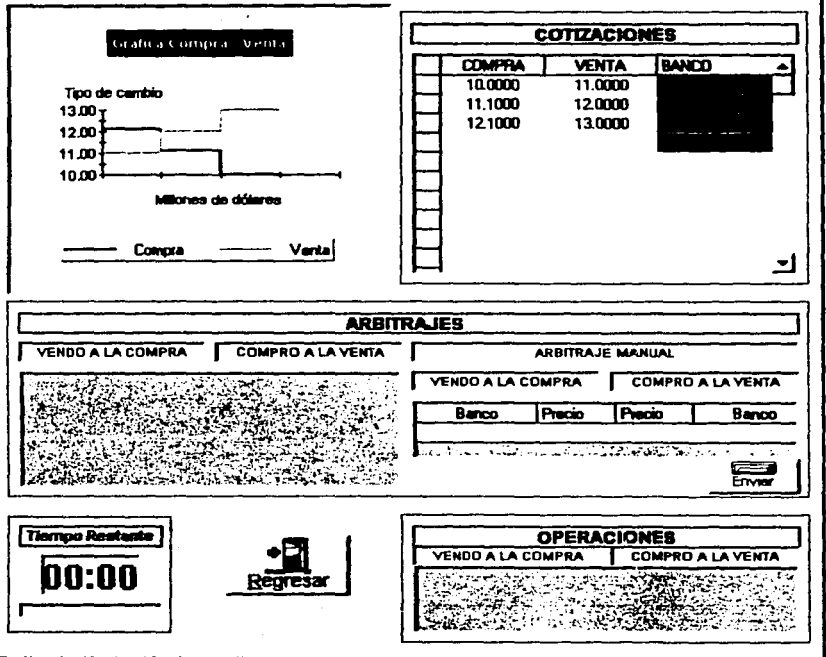


Figura 4.15 Estado en espera de clientes.

Una vez que se han recibido todas las posturas se procede al cálculo del FIX. La figura 4.16 muestra la pantalla de información que la sesión ha terminado.

TESIS CON FALLA DE ORIGEN

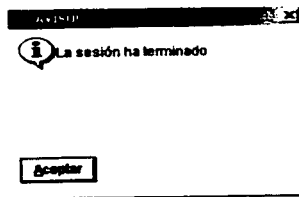


Figura 4.16 Fin de la sesión.

La figura 4.17 muestra los resultados de la primera sesión del día o el histórico de la primera sesión.

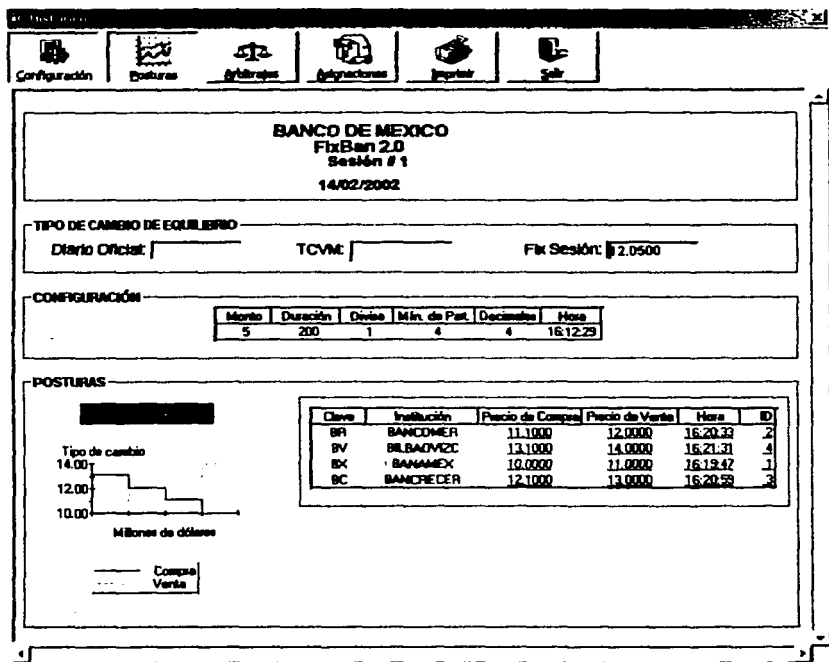


Figura 4.17 Histórico de la primera sesión del día.

Para realizar la segunda sesión del día se deben elegir al menos cuatro participantes distintos a los de la sesión previa. La figura 4.18 muestra los participantes elegidos para la segunda corrida.

El área de resultados del FIX muestra el resultado de la primera subasta.

Inicio de la segunda subasta

FIX

Sesión 1 12.0500 <input type="button" value="abrir"/>	Sesión 2 <input type="button" value="abrir"/>	Sesión 3 <input type="button" value="abrir"/>	Diario Of. <input type="button" value="abrir"/>	TCVM <input type="button" value="abrir"/>
MONTO (MIL. Dls.)		SESION ACTIVA		FK DA 12.9000
				TCVM DA 3.6132

BANCO

Código de Bancos		Permisos	Bancos Con Permisos
1 ABNAMRO 2 ATLANTIC 3 BITAL 4 BNP 5 BAWBA 6 CHASE 7 CITIBANK 8 DRESDNER 9 INBURSA 10 INDUSTRI 11 INO 12 JPMORGAN	13 MERCANO 14 NIFEL 15 NAFIN 16 REPUBLIC 17 BERFIN	<input type="button" value="abrir"/> <input type="button" value="abrir"/> Permisos: <input type="text" value="4"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> 1 INTERACC 1 INVERLAT 1 INMEX 1 DEX

BANCO 170.70.8.58 FIBan 2.0 Sesión # 1 Sesión activa 8 04:22 PM

Figura 4.18 Inicio de la segunda subasta.

La figura 4.19 muestra las posturas para la segunda sesión. El área de arbitrajes contiene una lista de los arbitrajes obtenidos con las últimas posturas.

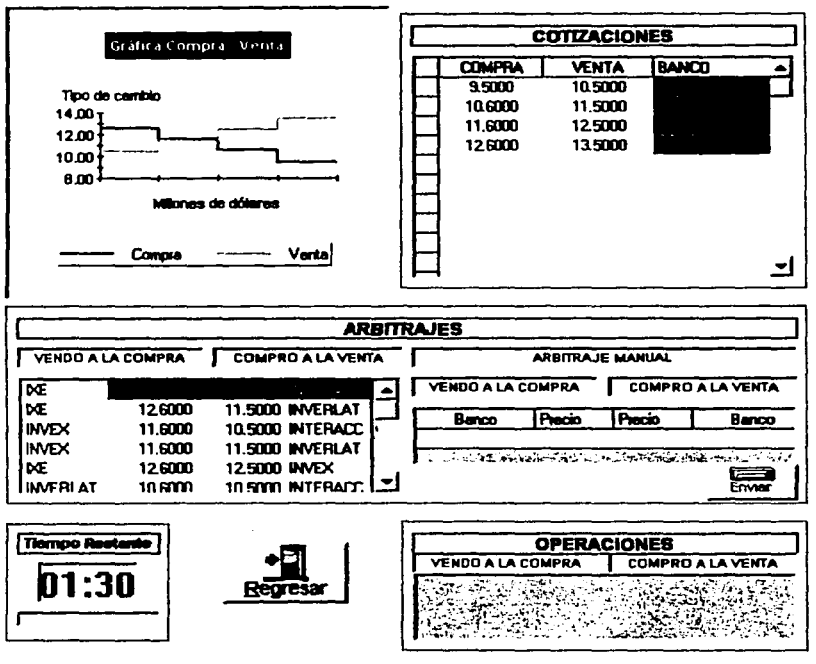


Figura 4.19 Arbitrajes de la segunda sesión.

Una vez que el reloj indique tiempo para arbitrajes, el operador puede seleccionar cualquier arbitraje (s) para la realización de asignaciones. La figura 4.20 muestra la pantalla que le permite escoger al operador el mejor arbitraje.

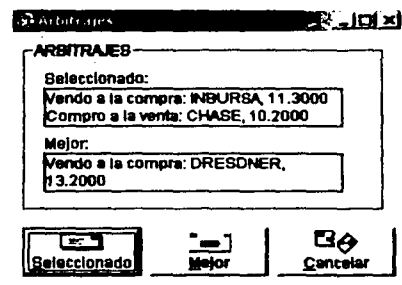


Figura 4.20 Selección del mejor arbitraje.

Una vez enviado el arbitraje, los bancos participantes en las asignaciones son notificados y la segunda subasta es terminada.

La tercera subasta implica la misma dinámica de las subastas anteriores. La figura 4.21 muestra los bancos que participaron en las tres subastas, los resultados del FIX de las tres sesiones, y el resultado del FIX y TCVM del día.

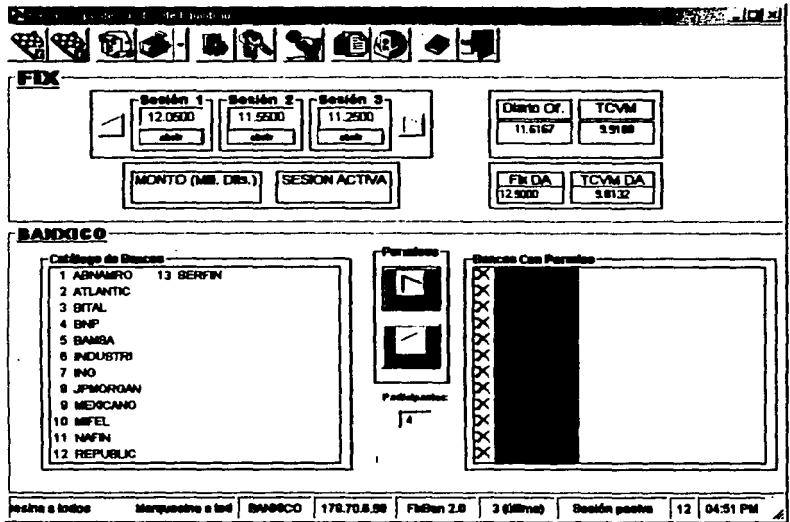


Figura 4.21 Resultados del FIX y TCVM del día.

El operador ahora puede generar las cartas para Comunicación Social y el comunicado para la Subgerencia de Cambios Nacionales, Figura 4.22.

TESIS CON
FALLA DE ORIGEN

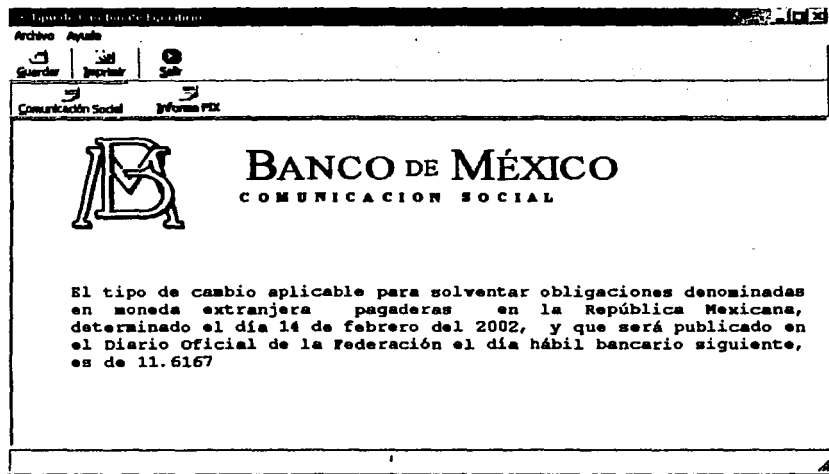


Figura 4.22 Carta para Comunicación Social.

También puede solicitar que se publiquen los resultados en Internet. Figura 4.23.

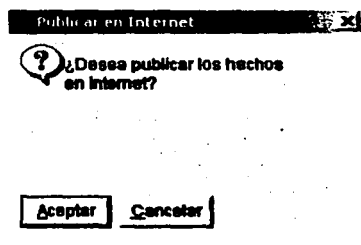


Figura 4.23 Publicación de los resultados en Internet.

TESIS CON FALLA DE ORIGEN

El operador puede acceder a los datos de las asignaciones obtenidas en el día. Figura 4.24.

ASIGNACIONES					
VENIDO A LA COMPRA			COMPRO A LA VENTA		
Sesión	Monto	Banco	Precio	Precio	Banco
2	5	DCE	12.6000	10.5000	INTERACC
2	5	INVEX	11.6000	11.5000	INVERLAT
3	5	INBURSA	11.3000	10.2000	CHASE

Figura 4.24 Asignaciones del día.

Capítulo 5

CONCLUSIONES

A continuación se destacan y sintetizan los principales hallazgos de este proyecto, los mismos que se sugieren sean considerados como pautas para el mantenimiento del sistema FixBan 2.0 cliente Banxico y el desarrollo de futuras tecnologías que lo reemplacen.

Se presentó el camino recorrido en la elaboración de un sistema de software. Desde el punto de partida los elementos teórico-prácticos aprendidos en la carrera de Ingeniería en Computación fueron las bases que guiaron en la selección de la ruta apropiada para llevar a un buen fin la conclusión del proyecto. La experiencia adquirida al elaborar éste proyecto no solo en la rama de la Ingeniería del Software, sino también en la de trabajo en equipo, representa una satisfacción invaluable.

Se optó por emplear los métodos estructurados en el análisis y el diseño debido a su simplicidad de uso y porque a la vez son ampliamente descriptivos. Se vio que no es necesario seguir al pie de la letra una metodología, sino ajustar sus herramientas según lo requiera cada situación dentro del ciclo de vida del proyecto.

La etapa de análisis se vio enriquecida con la buena disposición de los usuarios al indicar las carencias del sistema actual, sin embargo, en el diseño se tuvo que respetar los límites impuestos por los encargados del mantenimiento del servicio.

Los modelos obtenidos en la etapa de diseño simplificaron de manera práctica la programación de los módulos, dando como resultado un sistema organizado en módulos fáciles de mantener.

Conclusiones a nivel objetivo

Los resultados de las pruebas mostraron que el sistema satisface los requerimientos especificados por el usuario.

Se comprobó que el nuevo sistema no es solo más fácil de usar, debido a las nuevas funciones de impresión y generación de reportes incorporadas, sino que además brinda un mayor detalle de los sucesos ocurridos durante la ejecución de cada subasta.

Se vio que el protocolo de comunicación del FEC empleado en el diseño, mejora el desempeño del sistema actual, debido a que se lleva un mejor control de los participantes, y protege la comunicación directa con el servidor. El FEC pese a ser un filtro no genera un retraso en las comunicaciones.

El nuevo sistema elimina la necesidad de ser modificado cada vez que las especificaciones de los datos de la configuración cambien.

Aunque se trató de seguir al pie los modelos del diseño, a lo largo de la implementación surgieron algunos cambios:

- Modificación de los límites de automatización para la selección de arbitrajes.
 - Incorporación de la tarea de comunicaciones con la tarea de atención al FEC.
1. Modificación de los límites de automatización para la selección de arbitrajes. En el modelo de implementación del usuario que corresponde a la selección de los arbitrajes, el diseño especifica que el operador debe ser capaz de seleccionar el mejor arbitraje, sin embargo, para dar soporte a esta tarea y por su delicadeza se implementó una función que evalúa el o los arbitrajes escogidos, en caso de no haber seleccionado el mejor arbitraje se indicará la existencia de uno mejor, y se da al operador la opción de enviar el arbitraje elegido o el mejor. Esto se pensó que podría ser un arma de dos filos debido al corto tiempo que hay para enviar los arbitrajes, el introducir una caja de diálogo adicional podría entorpecer la operación. Sin embargo, se evaluó la importancia que significa someter el mejor arbitraje y se decidió por agregar la función.
 2. Incorporación de la tarea de comunicaciones con la tarea de atención al FEC. Aunque el diseño de tareas identificó tres tareas independientes: la de comunicaciones, la de atención al operador y la de atención al FEC, durante la implementación se optó por incorporar la tarea de comunicaciones y la del FEC en una, lo que facilitó la programación y se ajustó con el protocolo de mensajes empleado.

Recomendaciones para mejorar el diseño

A lo largo del desarrollo del sistema se observaron varias propiedades que el sistema podría agregar para mejorar su eficiencia:

- Proteger el cambio de la configuración de la subasta.
 - Agregar un módulo de recuperación de la sesión.
 - Incorporar el uso de tecnología de llave privada para la protección de los mensajes.
1. Proteger el cambio de la configuración de la subasta. Al permitir que la configuración de la subasta pueda ser cambiada se otorga al sistema flexibilidad para satisfacer futuros cambios en los rangos de los valores de los datos. Sin embargo, actualmente el cambio de la configuración puede ser realizado por el operador o cualquiera que tenga acceso al cliente Banxico, lo cual es indeseable. Es necesario agregar un módulo que autentifique al personal autorizado para el cambio de la configuración. Dicho módulo puede ser implementado a través de distintas tecnologías de autenticación, tales como: dispositivos físicos, entre los cuales se encuentran las llaves y tarjetas electrónicas, dispositivos lógicos, como el uso de passwords y frases secretas, y dispositivos biométricos, como la verificación de huellas digitales. Se sugiere la autenticación a través de password ya que su implementación causaría cambios menores al sistema.
 2. Agregar un módulo de recuperación de la sesión. Se discutió con el equipo de desarrollo del sistema completo sobre la importancia de un módulo de recuperación de la sesión, que fuera capaz de proveer de un respaldo lo suficientemente detallado como para continuar una

subasta justo en el último punto donde se quedó en caso de alguna falla en las comunicaciones durante una sesión. Este módulo se debe implementar en la parte del servidor debido a que está bajo el control de los ingenieros de mantenimiento, dejarlo en la parte del cliente podría dar la posibilidad de ser vulnerable a cambios.

3. Incorporar el uso de tecnología de llave privada para la protección de los mensajes. Esta propiedad está actualmente bajo desarrollo en la Subgerencia de Infraestructura Informática en la parte que corresponde al protocolo de comunicación. Su incorporación tanto en el cliente como en el servidor del FixBan requerirá posiblemente solo del cambio de la interfaz empleada en la lectura de los mensajes, ya que se busca hacerlo transparente a los sistemas que utilizan actualmente el protocolo de comunicación del FEC.

Para terminar queremos agregar que aunque generalmente el mérito de los sistemas de software se juzga exclusivamente por sus resultados, los beneficios que ofrecen son intangibles y difíciles de cuantificar, tanto para el cliente, el usuario y los ingenieros que participaron en su desarrollo.

BIBLIOGRAFÍA

- [Cooper 1995] Alan Cooper, *About Face: The Essentials of User Interface Design*, IDG Books Worldwide, Inc., 1995.
- [Diario Oficial 1996] Diario Oficial de la Federación, viernes 22 de marzo de 1996, pág. 9.
- [Dwayne Phillips 2000] D. Phillips, *The Software Project Manager's Handbook*, IEEE Computer Society, Los Alamitos CA, 2000, pág 148.
- [Hernández 1999] C. Hernández, *Front End de Comunicaciones*, Dirección General de Operaciones de Banca Central, Dirección de Sistemas Operativos y Cambios, Subgerencia de Infraestructura Informática, Banco de México, junio de 1999.
- [Meilir 1980] Meilir Page - Jones, *The Practical Guide To Structured Systems Design*, Yourdon Press, USA, 1980.
- [Yourdon 1989] E. Yourdon, *Modern Structured Analysis*, Yourdon Press, Englewood Cliffs, N.J., 1989.
- [Wymore 1993] A. W. Wymore, *Model-Based Systems Engineering*, CRC Press, Boca Raton, FL, USA, 1993.

Apéndice A

DICCIONARIO DE DATOS

- arbitraje* = *Se da cuando una postura de compra es mayor que una postura de venta en una subasta*
- arbitrajes* = {arbitraje}
- arbitraje-asignación*
= *Arbitraje seleccionado por el operador para realizar una asignación*
- asignación-válida*
= *Forma un arbitraje y se lleva a cabo dentro del tiempo de arbitrajes *

ASIGNACIONES

- = *Archivo donde se guardarán las asignaciones*
- asignación* = *Se lleva a cabo cuando el operador decide ejecutar las operaciones de compra y venta en un arbitraje, durante una subasta*
- asignaciones* = {asignación}
- banco* = *Nombre del banco que está inscrito para participar en el cálculo del FIX*
- cartas* = carta-com-soc / carta-comunicado
- carta-comsoc* = *Carta Comunicación Social*
- carta-comunicado*
= *Carta Comunicado*

COMUNICACION-SOCIAL

= *Archivo donde se guardará la carta de Comunicación Social*

COMUNICADOS

= *Archivo donde se guardará la carta Comunicado*

configuración-inicial

= *Parámetros enviados por el FEC, con los que se inicia el sistema*
lista-bancos + configuración-subasta + núm-sesión + estado-sesión +
{lista-estado-conexión-banco} + (posturas) + (arbitrajes) + (participantes) +
(resultados-FIX) + resultados-FIX-día-anterior

configuración-subasta

= *Parámetros para la configuración de las subastas*
num-sesiones + monto + duración + núm-participantes + decimales +
divisa

configuración-válida

= *Es la configuración de la subasta cuando todos los parámetros caen dentro del rango preestablecido*

datos-acceso

= *Información provista por el operador, necesaria para entrar al sistema*
passwd + dirIP-FEC

datos-cambio-passwd

= *Datos provistos por el operador para el cambio de password*
passwd-actual + passwd-nuevo

datos-histórico

= configuración-subasta + posturas + (arbitrajes) +
(asignaciones) + resultados-FIX

datos-indicaciones

= *Indicaciones en forma de mensaje o marquesina que el operador solicita se envíe a bancos específicos.
{banco} + mensaje | marquesina + texto

decimales

= *Parámetro de la subasta, que establece el número de dígitos decimales considerados en el cálculo del FIX*
4 ≤ decimales ≤ 8

decimales-válidos

= *El valor del parámetro decimales que cae dentro del rango especificado*

desconexión

= *Mensaje generado cuando el FEC no responde a los mensajes de comunicación*

dirIP-FEC

= *Dirección Internet numérica que corresponde a la máquina donde corre el FEC*

diálogo FEC

= *La comunicación bidireccional entre el sistema y el FEC*.

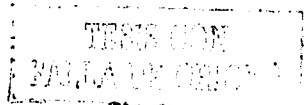
diálogo-operador

= *Las instrucciones que el operador le da al sistema y los mensajes de respuesta que el sistema regresa al operador*.

divisa

= *Valores subastados*
Unidad: dólar

- divisa-válida* = *El valor del parámetro *divisa* verificado como válido*
- duración* = *Parámetro de la subasta, que establece su duración*
 Unidades: segundos
 $30 \leq \text{duración} \leq 70$
- duración-válida* = *El parámetro *duración* que cae dentro del rango establecido*
- estado-conexión-banco* = *Información enviada por el FEC para notificar cuando un banco se ha conectado o desconectado*
 [conectado | desconectado]
- estado-sesión* = *Estado de la sesión actual, o último estado en el que se quedó la sesión si ya había sido empezada*
 [pasiva | en-tiempo | en-arbitrajes | esperando-clientes]
- estas-vivo?* = *Mensaje que envía el FEC preguntando si el cliente Banxico permanece conectado*
- FEC* = *Front End de Comunicaciones, programa que funciona como interfaz entre los clientes y el servidor*
- fin-arbitrajes* = *Mensaje que indica que el tiempo de arbitrajes ha terminado*
- fin-posturas* = *Mensaje que indica que el tiempo para aceptación de posturas ha terminado*
- fin-sesión* = *Mensaje que indica que la sesión ha terminado*
- fin-programa* = *Mensaje que indica que el programa terminará*
- FIX* = *El resultado del cálculo del tipo de cambio aplicable para solventar obligaciones denominadas en moneda extranjera pagaderas en la República Mexicana*
- FIX-día* = *Promedio de los resultados del FIX de las sesiones en el día*
- FIX-sesión* = *Resultado del cálculo del FIX en la sesión*
 ID-sesión + FIX
- FORMATO COMUNICACION SOCIAL**
 = *Archivo que contiene el formato para la carta de Comunicación Social*



FORMATO COMUNICADO

= *Archivo que contiene el formato para la carta Comunicado*

formato-comsoc

= *Datos para generar la carta de Comunicación Social*

formato-comunicado

= *Datos para generar la carta Comunicado*

gráfica-posturas

= *Gráfica formada por las últimas posturas de los participantes*

HISTORICOS= *Archivo donde se guardará el histórico*

histórico-sesión

= *Datos del histórico presentados al operador en forma de reporte*

id-error

= *Identifica el error generado*

ID-sesión

= *Identificador de la sesión*

inicio-subasta

= *Mensaje de indicación de inicio de la subasta*

inicio-arbitrajes

= *Mensaje que indica que empieza el tiempo para arbitrajes*

lista-bancos

= {banco}

LISTA-PARTICIPANTES

= *Archivo que contiene la lista de participantes*
{participante}

lista-participantes

= {participantes}

lista-estado-conexión-bancos

= {estado-conexión-banco}

login

= *Nombre que identifica al operador en el sistema*
Valor: "banxico"

longitud-permitida

= *Longitud del password*
Unidades: caracteres
5 ≤ longitud ≤ 8

- marquesina* = *Información corta enviada por el operador hacia una selección de bancos*
longitud-marquesina < 50 caracteres
- mensaje* = *Información enviada por el operador hacia una selección de bancos*
- monto* = *Parámetro de la subasta que especifica la cantidad de divisas a subastar*
Unidades: dólar
0 < monto
- monto-válido* = *El valor del parámetro *monto* que cae dentro del rango especificado*
- nombre-archivo* = *Nombre del archivo donde se guardará el reporte*
- núm-histórico* = *Número para identificar el histórico de la sesión que el operador solicita*
- núm-participantes* = *Parámetro de la subasta que especifica el número mínimo de bancos necesarios para realizar la subasta*
4 ≤ *núm-participantes*
- núm-participantes-válido* = *El valor del parámetro *núm-participantes* que cae dentro del rango especificado*
- núm-sesión* = *Número de la última sesión en el día*
- núm-sesiones* = *Parámetro de la subasta que especifica el número de sesiones que se realizarán en el día*
3 ≤ *núm-sesiones*
- núm-sesiones-válida* = *El valor del parámetro *núm-sesiones* que cae dentro del rango especificado*
- operador* = *Personal de la oficina de Mercado de Valores que tiene permisos para operar el sistema*
- participantes* = *Bancos seleccionados por el operador para participar en una subasta*
- participantes-inválidos* = *Son inválidos si ya participaron en una subasta previa durante el

mismo día*

participantes-válidos

= *No han participado previamente en alguna subasta del mismo día*

password

= *Clave secreta usada para autenticar al operador*
{A-Z} + {a-z} | {0-9} | caracter-especial | {0-9} + caracter-especial

passwd

= *Alias de password*

passwd-actual

= *Password actual*

passwd-actual-auténtico

= *passwd-actual auténtico*

passwd-fuerte

= *Password compuesto por letras mayúsculas, minúsculas y números o caracteres especiales*

passwd-nuevo

= *Password nuevo*

passwd-nuevo-válido

= *Es el passwd-nuevo que cumple con los requerimientos de seguridad preestablecidos*

postura

= *El precio de compra y el precio de venta que los bancos participantes someten durante la subasta*

posturas

= {postura}

precio-compra

= *Precio que ofrece pagar el banco participante a la compra*

precio-venta

= *Precio que solicita el banco participante a la venta*

protocolo-acceso

= *Reglas preestablecidas que especifican el intercambio de datos necesarios entre el cliente FIXBan 2.0 y el FEC, en el proceso de acceso al sistema*

solicitud-conexión →	← respuesta-conexión
solicitud-autenticación →	← respuesta-autenticación
versión →	← respuesta-versión
	← configuración-inicial

puerto

= *Punto de conexión preestablecido por el administrador, donde se buscará establecer comunicación con el FEC*
7056 ≤ puerto ≤ 7060

REPORTES = *Archivo donde se almacenan los resultados para reportes*.

reporte-asignaciones

= *Reporte de todas las asignaciones ocurridas en el día*

respuesta-acceso

= *Respuesta obtenida cuando el operador solicita entrar al sistema*
[éxito | error]

respuesta-asignación

= *Respuesta enviada por el FEC indicando si aceptó o no la asignación*
[éxito | error]

respuesta-autenticación

= *Respuesta que indica el resultado de la autenticidad del operador*
[éxito | error]

respuesta-cambio-password

= *Resultado obtenido cuando el operador solicita cambio de password*
[éxito | error]

respuesta-cambio-configuración

= *Respuesta enviada al operador cuando solicita cambio de configuración de la subasta*
[éxito | error]

respuesta-conexión

= *Respuesta que indica si se pudo establecer conexión con el FEC*
[éxito | error]

respuesta-configuración

= *Respuesta enviada por el FEC para indicar si actualizó la configuración de la subasta*
[éxito | error]

respuesta-guardar

= [éxito | error]

respuesta-impresión

= [éxito | error]

respuesta-password-actual

= *Respuesta enviada por el FEC indicando si el password actual es auténtico*
[éxito | error]

- respuesta-password-nuevo*
= *Respuesta enviada por el FEC para indicar si aceptó el cambio de password*
[éxito | error]
- respuesta-publicar-Internet*
= *Respuesta enviada por el FEC indicando si se pudo publicar el resultado del FIX del día en Internet*
[éxito | error]
- respuesta-versión*
= *Respuesta que indica si las versiones del cliente y el servidor coinciden*
[éxito | error]
- resultados-FIX* = (FIX-día) + (TCVM-día)
- resultados-FIX-día-anterior*
= *resultados-FIX del día hábil previo*
- selección-asignaciones*
= *Selección del reporte de las asignaciones del día*
- selección-comunicación-social*
= *Selección de la carta para Comunicación Social*
- selección-comunicado*
= *Selección de la carta Comunicado*
- selección-histórico*
= *Selección del reporte del histórico de alguna sesión*
- servidor*
= *Nombre del servidor*
Valor: "FIXBan 2.0"
- servidor-no-disponible*
= *Respuesta enviada por el FEC indicando que el servidor no está activo*
- sesión-en-arbitrajes*
= *Abarca los 10 segundos que le siguen a la duración de la subasta*
- sesión-en-tiempo*
= *Estado de la subasta en el que se reciben las posturas de compra-venta de los bancos. Se está "en-tiempo" durante los segundos especificados en duración*

- sesión-esperando-clientes* = *Se da al *terminar* el tiempo de arbitrajes y solo cuando no se han recibido al menos una postura por cada *participante**
- sesión-pasiva* = *Estado anterior o posterior a la ejecución completa de una subasta*
- solicitud-autenticación*
= *Solicitud de autenticación para conectarse con el servidor FixBan*
login + paswd + servidor
- solicitud-cartas* = *Petición de la generación de las cartas hecha por el operador*
- solicitud-conexión* = *Petición para establecer conexión con el FEC en una máquina remota*
- solicitud-guardar*
= *Solicitud del operador del guardado de algún reporte*
- solicitud-imprimir*
= *Solicitud del operador de la impresión de algún reporte*
- solicitud-inicio-subasta*
= *Petición de ejecución de una subasta*
- solicitud-publicar-Internet*
= *Solicitud para enviar los resultados del FIX del día a Internet*
- solicitud-resultados-asignaciones*
= *Petición de todas las asignaciones ocurridas en el día*
- subasta* = *Estado de la sesión en la que se pone a la venta un monto específico de divisas (dólares). Mientras dura la subasta los participantes deben someter sus posturas de compra y venta. Estas posturas se utilizan para realizar el cálculo del valor del FIX*
- tabla-arbitrajes* = *Muestra los posibles arbitrajes de la subasta*
- TCVM-día* = *Tipo de Cambio del Valor Medio*
- tiempo-sesión* = *Avance del tiempo de la sesión enviado por el FEC*
- tipo-documento*
= *Tipo de reporte a ser guardado*
- vivo* = *Mensaje enviado por el sistema requerido para mantener comunicación con el FEC*
- versión* = *Versión actual del FIXBan*
Valor: "2.0"