

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

95
2EJ



DESARROLLO DE UN SISTEMA DE MONITOREO
DE MERCADOS UTILIZANDO ARQUITECTURA
CLIENTE / SERVIDOR

FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A
GINES RIVERO GARCIA

MEXICO, D. F.

1995



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.

A mis padres, Ginés Rivero García y Flavia García de Rivero por todo el apoyo que me dieron siempre.



Tabla de Contenido

INTRODUCCION	1
OBJETIVOS	3
FORMA CONVENCIONAL	5
METODOLOGIA INCREMENTAL	7
ARQUITECTURA DE APLICACIONES CLIENTE/SERVIDOR	9
ENTORNO DEL NEGOCIO	15
ANALISIS	19
MODELO LOGICO DE DATOS	22
MODELO FISICO DE DATOS	24
DISEÑO CONCEPTUAL DE LA APLICACION	29
CONFIGURACION DE AMBIENTE	29
ESTANDARES Y CONVENIOS	30
ORGANIZACION DE OBJETOS	35
LOGISTICA DE LA APLICACION	37
DESARROLLO DE LA APLICACION	40
CONSTRUCCION	41
DISEÑO DE PANTALLAS	44
PRUEBAS	69
PUESTA EN MARCHA DE LA APLICACION	71
CREACION DEL PROGRAMA EJECUTABLE	71
DISTRIBUCION DE LA APLICACION A USUARIOS FINALES	73
CONCLUSIONES	76
BIBLIOGRAFIA	78
GLOSARIO DE TERMINOS FINANCIEROS	79
GLOSARIO DE TERMINOS TECNICOS	82



Introducción

Antecedentes

Debido a que la industria del Procesamiento de Información cada vez tiene un mayor número de demandantes en nuestro país y en el mundo, es necesario estar preparados para poder tener un alto nivel de competitividad. Desafortunadamente los sistemas no se están desarrollando en el tiempo estimado y los que se llegan a desarrollar son inadecuados.

Ya que la información es un punto crítico en cualquier empresa de esta década, tenemos un fuerte reto que afrontar para poder cubrir las necesidades actuales y así poder competir dentro del mercado, para ello se requiere la utilización de nuevos conceptos y tecnología de punta.

Las formas convencionales utilizadas para el desarrollo de sistemas no están teniendo los resultados esperados. Es verdad que algunos de los sistemas desarrollados han tenido éxito, pero las estadísticas indican claramente que, nosotros como industria, estamos fallando. Podemos citar un ejemplo de un estudio que se hizo en el gobierno de los Estados Unidos con respecto a nueve contratos de software costando 6.8 millones de dólares, arrojando resultados nada halagadores. Solo el 2% de los sistemas fueron utilizados al ser liberados. El 3% fueron utilizados después de que se realizaron cambios. Esto significa que solo el 5% de los sistemas fueron realmente utilizados. Para mayor información (Boehm B. "Software Engineering Economics, Prentice Hall 1981; Blum, B. "Understanding the Software Paradox". "ACM Software Engineering Notes" 10/1 1985).

Para los sistemas que han sido instrumentados satisfactoriamente, las estadísticas nos ponen a meditar, ya que de acuerdo con James Martin y Carma Mc Clure en su libro "Software Maintenance: The Problem and its Solution", el 67% del costo de un sistema ocurre en las fases de Operación y Mantenimiento dentro del ciclo de vida de un sistema.

Un extracto del libro de James Martin, "An Information System Manifesto" publicado en 1984, indica que "la mayoría de las corporaciones tienen pendientes que van de 2 a 4 años". En otras palabras, toma de 2 a 4 años para desarrollar sistemas de acuerdo a las necesidades del usuario. Aun más, muchos usuarios simplemente no hacen requerimientos porque saben que tomará mucho tiempo para que vean los resultados y en ocasiones ni siquiera los llegan a ver.

Estas estadísticas indican ineficiencia en el desarrollo de sistemas utilizando formas convencionales, pero lo peor es que nosotros como industria estamos aceptando este nivel tan bajo de productividad y calidad y si



Monitor de Mercados

queremos sobrevivir al nuevo siglo tendremos que mejorar la calidad de nuestras soluciones propuestas como consultores.

La tendencia actual de la industria es producir aplicaciones modelos que describan y apoyen a la empresa, verdaderamente realizar negocio, con la presión que se tiene de la globalización económica a nivel mundial y el incremento de la competencia, nuestros clientes requieren información rápida y confiable que les permita a ellos tomar decisiones que les aseguren la sobrevivencia y éxito en sus diferentes mercados.



Objetivos

- Proporcionar al usuario una aplicación que su interfase sea transparente y amigable con respecto al medio ambiente de trabajo actual del usuario, sin sacrificar por ninguna la razón la funcionalidad de la misma.
- Ser utilizada como una herramienta de soporte informativo para cada uno de los usuarios que la utilicen y que la información mostrada se pueda utilizar para tomar decisiones importantes en determinados momentos.
- Incrementar la productividad a lo largo del desarrollo de la aplicación a través del lenguaje de cuarta generación utilizado (PowerBuilder), logrando disminuir por consiguiente el tiempo en el desarrollo de la misma.
- Involucrar de manera total al usuario en el desarrollo del sistema, de tal forma que se pueda romper la barrera que existe entre la gente que desarrolla la aplicación (sistemas) y la gente que utilizará la aplicación (negocio).
- Eliminar rentas de productos de información externos (INFOSEL) provocando de esta manera la reducción de costos.
- Eliminar el tiempo que el usuario pierde al tener que imprimir información de la TANDEM o INFOSEL para su análisis, ya que con la aplicación cada usuario tendrá la aplicación en su PC.
- Aprovechar el medio ambiente de trabajo del usuario (Windows) para disminuir la curva de aprendizaje que representa el trabajar con una nueva aplicación, de tal forma que el usuario vea la aplicación desarrollada como un icono más dentro de su computadora.
- Utilizar la información que la aplicación les proporciona, pudiendo realizar análisis sobre ella para una posible transformación
- Ser el inicio de futuras aplicaciones dentro de la misma arquitectura de cliente/servidor, utilizando las mismas plataformas (Intel, Unix), base de datos (Sybase System 10) y lenguaje de cuarta generación (PowerBuilder) aprovechando la infraestructura ya existente.



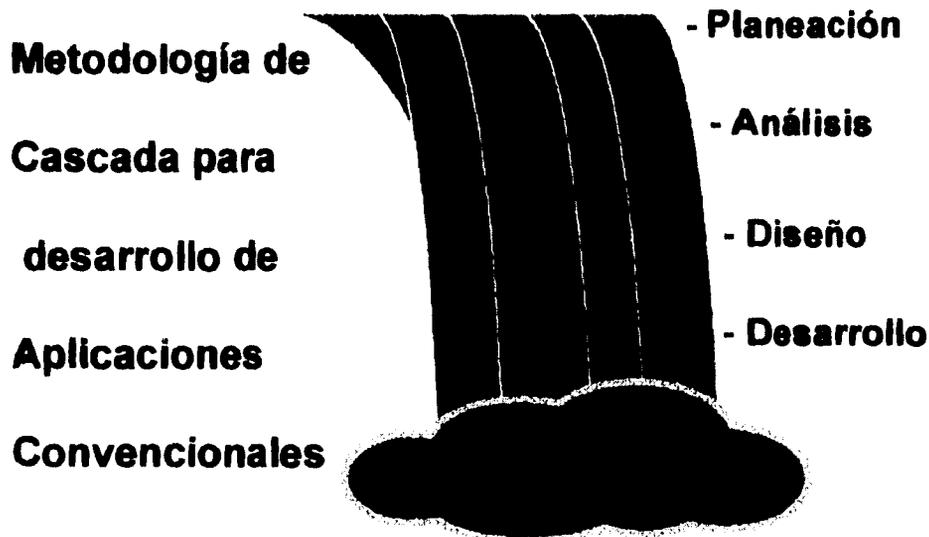
Monitor de Mercados

- **Adquirir experiencia en el desarrollo de aplicaciones de este tipo, aprovechando las herramientas utilizadas en esta aplicación, para que de esta manera se tenga una posición tecnológica privilegiada con respecto al mercado.**



Forma Convencional

En los proyectos para desarrollo de aplicaciones en mainframe, tradicionalmente se maneja la metodología de "cascada" en el mejor de los casos, en la que primero se realiza una etapa de planeación, al terminar esta etapa se comienza la etapa de análisis. Después del análisis, se comienza la etapa de diseño y posteriormente la etapa de desarrollo o construcción, instrumentación, etc. Con este modelo es muy costoso regresar a un paso hacia atrás, debido a que el tiempo invertido entre etapa y etapa es muy grande.



Esta metodología es esencialmente lineal iniciando en la parte de arriba con la planeación cayendo hasta la implementación del sistema. La gente de sistemas involucrada en estos proyectos trata de obtener firmas de los usuarios de alto nivel, como una forma de estar de acuerdo con el trabajo realizado en cada una de estas etapas.

Desgraciadamente para cuando se terminó el desarrollo o la construcción del sistema, los usuarios que firmaron alguna de las etapas anteriores ya no trabajan en la compañía, como este proceso se lleva entre dos y tres años, el sistema terminado trabaja con requerimientos obsoletos convirtiéndose en un sistema no funcional, que no cumple con las necesidades actuales del negocio.

Los sistemas que logran ser terminados utilizando esta metodología, cuentan con pantallas de caracteres, forzando al usuario a navegar a través de la aplicación utilizando menús, teclas de función, y opciones



Monitor de Mercados

limitadas, haciendo al sistema nada amigable, ya que para llegar a la pantalla deseada se tiene que pasar algunas veces hasta por 7 pantallas, si el usuario no se sabe la tecla de función rápida para llegar a ella.

Es importante considerar que la metodología tradicional no fue creada para soportar las nuevas tecnologías basadas en GUI's , orientación a objetos (object-oriented) y desarrollo rápido de aplicaciones (Rapid Application Development RAD), es un hecho que la mayoría de las herramientas y metodologías que se utilizan hoy en día no existían cuando fue creada la metodología tradicional.

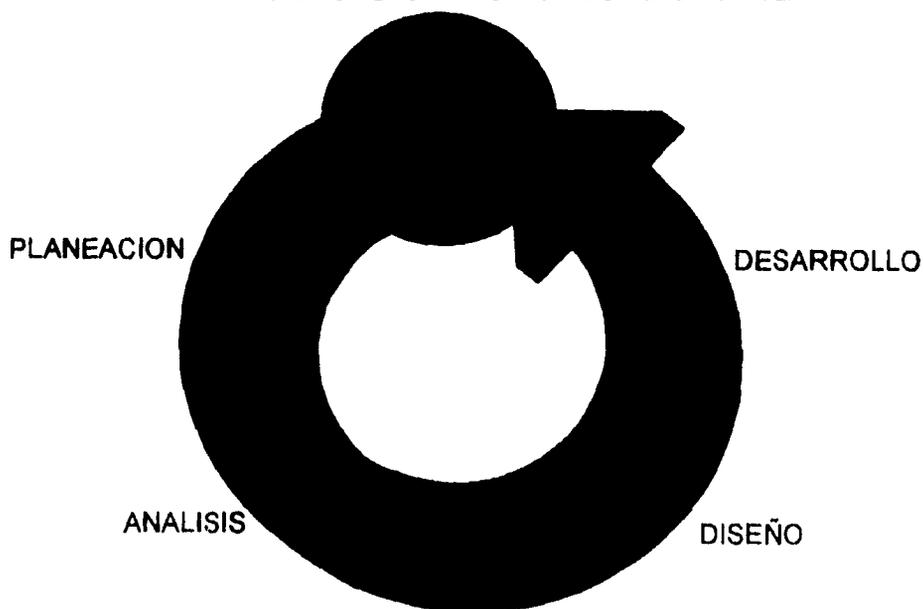
Años y años de desarrollo de sistemas inadecuados dieron como resultado un largo mantenimiento y retroceso tecnológico (promediando cuatro años en una típica compañía de los ochentas) ayudando con ello, a la precipitada era de las computadora de usuario final (computing end-user era). Esta era enfocada al usuario final permitió a la arquitectura cliente/servidor, desarrollar herramientas en la pc, y como resultado de esta evolución, los usuarios de hoy en día demandan estar involucrados en el proceso de desarrollo de cualquier sistema o aplicación.



Metodología Incremental

Para la arquitectura cliente/servidor se utiliza el modelo de "espiral" para desarrollar sistemas de manera incremental. En el modelo incremental se manejan ciclos, en donde en el primer ciclo se pueden identificar varias funciones comunes, contemplarlas dentro de la planeación, realizar análisis, diseño y desarrollo de un prototipo. En este punto se puede decidir si se debe continuar con un nuevo ciclo de desarrollo o cancelar el sistema antes de que se invierta más tiempo, dinero, análisis y diseño. Cerca del 50% de todos los sistemas a desarrollar son cancelados en alguna de estas fases dentro del primer ciclo.

PRIMERA ITERACION UTILIZANDO METODOLOGIA INCREMENTAL



Utilizando este tipo de modelo en el siguiente ciclo se pueden agregar más funciones, volver a planear, diseñar, desarrollar y nuevamente decidir si se continúa. En la medida que se incrementa el desarrollo, el usuario puede ver de manera tangible lo que se lleva de desarrollo (prototipo) y puede decirnos si lo encuentra funcional y si realmente así lo esperaba. Muchas de las veces los usuarios se dan cuenta del potencial de un sistema y del provecho que le pueden sacar a éste, hasta que lo empiezan a utilizar. El problema es que con los métodos tradicionales no siempre tenemos la fortuna de mostrarle a los usuarios lo que hubiera sido su sistema



Este modelo permite que un sistema pueda ir creciendo en la medida en que las necesidades del negocio cambian, haciendo al sistema fácilmente mantenible y desarrollable, ya que no solo se podrán ir corrigiendo bugs, sino que también se podrán realizar cambios que cubran las directrices del negocio de nuestros clientes. El modelo de espiral utiliza la metodología de rápido Desarrollo de aplicaciones (Rapid Application Development RAD) así como conceptos de la metodología Joint Application Development JAD.

Con la metodología incremental se pueden iniciar proyectos, diseñar modelos lógicos de datos, prototipos, etc, prosiguiendo con este proceso en ciclos, es posible que entre más vueltas se de al espiral, se va llegar a un estado tal que los productos que se agreguen al sistema, van a ir disminuyendo, debido a que se ha dado un refinamiento total del sistema, desechando lo que ya no se utiliza.

En efecto un sistema bajo la arquitectura cliente/servidor y manejando metodologías incrementales puede nacer, crecer e ir evolucionando dependiendo de los cambios que se vayan presentando en el negocio al cual esta enfocado, sin convertirse en un sistema estático, a punto de extinguirse.

Arquitectura de Aplicaciones Cliente/Servidor

Los cambios en la tecnología de información ha provocado que la arquitectura Cliente/Servidor sea la preferida para proporcionar servicios de información que cubran las expectativas de nuestros clientes hoy en día.

Los usuarios de hoy demandan sistemas que sean fáciles de usar, que realmente sean funcionales, viendo estos como una verdadera herramienta de trabajo. La arquitectura Cliente/Servidor proporciona beneficios y facilidades para cubrir este tipo de expectativas.

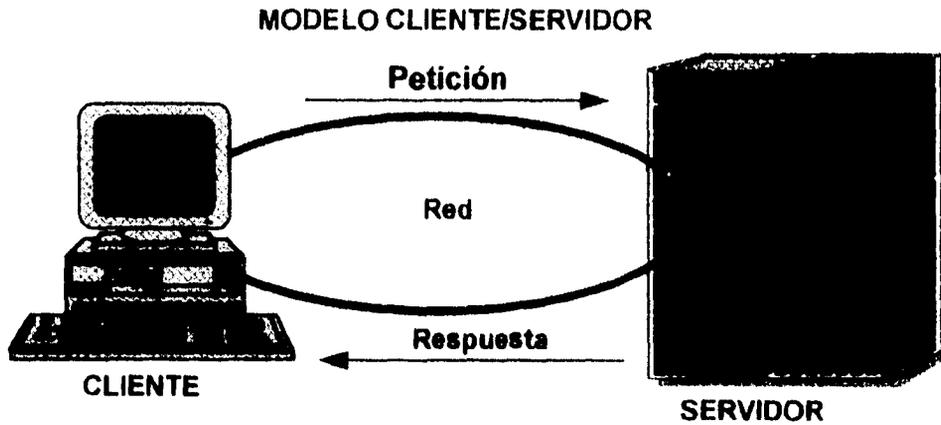
La arquitectura Cliente/Servidor es un procesamiento computarizado caracterizado por la división de aplicaciones en componentes. El termino Cliente/Servidor se refiere a que una aplicación o nodo, conocida como Cliente, y otra aplicación o nodo, conocida como Servidor, trabajan y se comunican a través de una red. La definición de cada uno de componentes de la arquitectura Cliente/Servidor es:

COMPONENTE	DEFINICION
Cliente	Una aplicación o proceso que realiza solicitudes de servicios o de información a otras aplicaciones o procesos (servidores)
Servidor	Una aplicación o proceso que procesa peticiones de clientes, brinda servicios, y regresa la información solicitada al cliente.
Red	Permite el intercambio de información entre el cliente y el servidor.

Las características de la arquitectura Cliente/Servidor son:

- Existen dos entidades totalmente distintas, la entidad solicitadora de servicios (Cliente) y la entidad proporcionadora de servicios (Servidor).
- Los componentes del cliente y el servidor pueden operar en plataformas separadas.
- Tanto el cliente como el servidor pueden tener crecimiento en forma independiente sin tener que crecer el otro componente.
- El servidor es capaz de atender múltiples clientes
- Los componentes del cliente y del servidor se comunican a través de una red

Adicionalmente a estas características, es posible tanto para el cliente como para el servidor, trabajar en forma independiente, realizando tareas fuera de una relación cliente/servidor dada



Un cliente realiza una petición de servicio o información a un servidor. El servidor procesa la petición y regresa la información al cliente

Para diseñar una aplicación cliente/servidor, es necesario primero dividir la aplicación en componentes funcionales. Los mayores tres componentes funcionales que puede tener una aplicación son:

COMPONENTE	FUNCIONES
Presentación o interfase de usuario	<ul style="list-style-type: none">• Presenta los servicios de la aplicación al usuario final.• Dirige el uso del teclado, mouse, para poder seleccionar objetos que manipulan información.• Incluye tanto interfaces gráficas como de texto.
Aplicación lógica o procedimientos enfocados al negocio	Manipulación de datos, generación de reportes y gráficas. Responsable de la implementación de funciones que requiere el negocio.
Manejo de Datos	Proporciona el acceso a datos, integridad de datos, seguridad de los datos, backups, recovery.

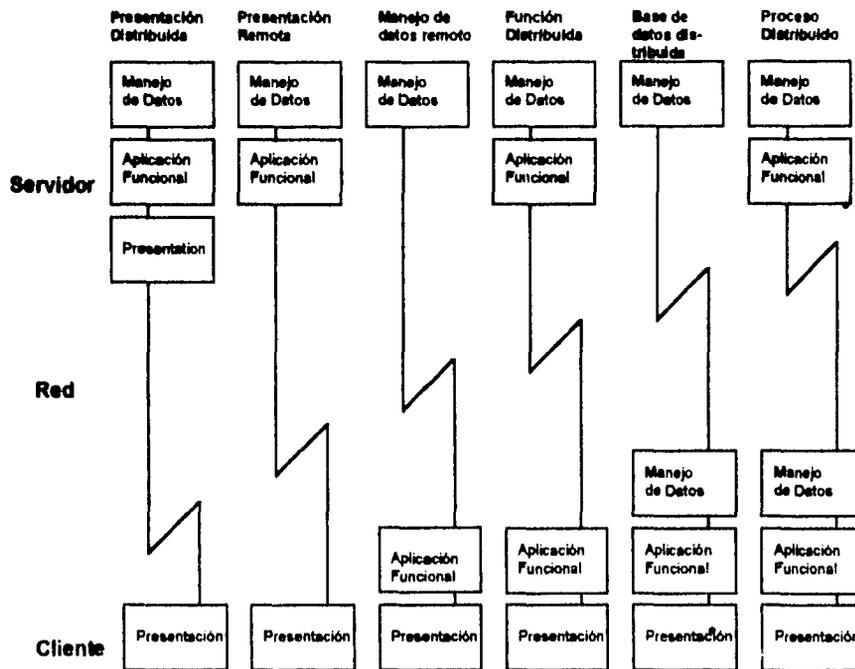


Estos componentes funcionales de una aplicación son los elementos básicos dentro de la arquitectura cliente/servidor.

Basándonos en el "Gartner Group's Model" para arquitecturas cliente/servidor, a continuación se presenta un modelo de alto nivel (ver figura) donde se manejan seis esquemas de las arquitecturas posibles de cliente/servidor :

- Presentación Distribuida
- Presentación Remota
- Manejo Remoto de Datos
- Función Distribuida
- Base de Datos Distribuida
- Procesos Distribuidos

ESQUEMAS DE ARQUITECTURA CLIENTE/SERVIDOR



Estos seis esquemas se distinguen uno del otro por la forma en que se encuentran colocados los componentes funcionales (presentación, aplicación funcional, manejo de datos) entre la plataforma del cliente y del



servidor. Va a depender de las necesidades del negocio y del costo operacional, la forma en que estos componentes sean colocados. A continuación se describen de manera general cada uno de estos esquemas

Presentación Distribuida

Esquema que representa el nivel más bajo dentro de la arquitectura cliente/servidor en donde el manejo de datos, las funciones de la aplicación y la presentación se encuentran en el servidor. La presentación que es la interfase con el usuario se encuentra también del lado del cliente.

Con este esquema los procesos que el cliente (desktop) realiza son mínimos, ya que solo realiza tareas como pintar la pantalla, detectar movimiento del mouse, alguna tecla oprimida, etc y mandar mensajes a través de la red para que desde el servidor se valide e identifique la(s) tarea(s) a realizar.

Presentación Remota

Toda la presentación se encuentra alojada en el cliente y lo demás (funciones de la aplicación y manejo de datos) se realiza del lado del servidor. Con este esquema cada tecla oprimida o movimiento del mouse no tiene que ir a través de la red, el cliente recoge y valida la entrada y crea mensajes de alto nivel o peticiones para enviarle al servidor. Comparando el esquema de Presentación Remota con el de Presentación distribuida, este último nos permite realizar algunos procedimientos desde el lado del cliente

Manejo de Datos Remoto

Con este esquema los datos y las funciones para acceder los datos se encuentran alojados en el servidor mientras que en el cliente se realizan los procedimientos de la aplicación así como operaciones con los datos obtenidos. Las funciones de presentación se encuentran también en el cliente.

Funciones Distribuidas

En el esquema de Funciones distribuidas algunas de las funciones de las aplicaciones corren del lado del cliente y otras corren del lado del servidor. La presentación corre del lado del cliente y el manejo de los datos del lado del servidor.

Bases de Datos Distribuidas

Parte de los datos están en el cliente y parte en el servidor. todas las funciones de la aplicación y la presentación están en el cliente. Este esquema es otra manera de resolver el trabajo de carga de la red, colocándose ciertos datos cerca de las funciones del lado del cliente.

Procesos Distribuidos



Este esquema tiene repartido los datos y las aplicaciones entre el servidor y el cliente. Las aplicaciones pueden ser diseñadas e instrumentadas de tal forma que cualquier función o dato puede ser movido a cualquier componente de la configuración.

Para el desarrollo de la presente aplicación, se utilizó el esquema de Funciones Distribuidas, considerando que el tiempo de respuesta que debía tener cada cliente debía ser alto, así como del lado del cliente se podrían manipular ciertos datos (filtros de información, ordenamientos, etc) una vez obtenidos, disminuyendo de esta manera el tráfico en la red.

Dentro del desarrollo de un sistema en la arquitectura cliente/servidor, se requiere que las personas que participan en este, sean especialistas en el (los) papel (es) que van a desempeñar. Para identificar mejor los diferentes papeles que se pueden desempeñar dentro del desarrollo de una aplicación, a continuación se presenta una tabla donde se indica el papel que puede tener un participante, lo que implica el hecho de representar ese papel y en el tipo de componente básico en el que éste trabajará.

PAPELES	LO QUE IMPLICA	COMPONENTES BASICOS
Supervisor	Tener un panorama general de la aplicación, tomar decisiones, calendarizar, coordinar	Conocimientos generales de los tres componentes (cliente, red, servidor)
Representante del Usuario Final	Entender el alcance de la aplicación, trabajar directamente con el diseñador.	Trabaja directamente en el cliente
Diseñador	Cubrir los requerimientos del usuario y que estos apliquen dentro del ambiente del sistema.	Trabaja tanto en el lado del cliente como en el del servidor.
Administrador de la base de datos	Responsable de las tablas que forman parte de la base de datos. Trabajar directamente con el diseñador	Trabaja en lado del servidor.
Administrador de la red	Configurar el ambiente de red, verificar que todos los clientes se encuentren conectados a los servidores.	Trabaja en la red y en el cliente.



Monitor de Mercados

Responsable del manejo de objetos	Administrar y controlar los diferentes objetos comunes creados desde Powerbuilder para poderlos rehusar en otras aplicaciones	Trabaja en el cliente
Desarrollador	Crear y dar mantenimiento a los objetos de Powerbuilder (implica dibujarlos, meterles código, manejar SQL's, además de estar familiarizado con el sistema operativo del cliente)	Trabaja tanto en el lado del cliente como en el del servidor.

Para este estudio, el material que se presenta en el desarrollo de la aplicación, es considerando los papeles de un diseñador, del responsable del manejo de objetos y del desarrollador y solo se mencionaran o se asumirán tareas que cualquiera de los otros papeles (administrador de la base de datos, administrador de la red, y gerente) hayan realizado, de lo contrario podríamos caer en el riesgo de hablar sin fundamento de algún tema.



Entorno del Negocio

Esta aplicación esta enfocada a aquellos usuarios que de alguna forma están relacionados con el manejo de operaciones que se realizan en la Bolsa Mexicana de Valores, para ello a continuación se hablará en forma general del sistema financiero mexicano, del mercado de valores que es parte del sistema, de sus divisiones y su estructura, todo esto con el fin de tener un mejor entendimiento de la información que la aplicación manejará.

Contexto General

El sistema financiero mexicano es un conjunto de instituciones que captan, administran, norman, regulan, orientan y dirigen tanto el ahorro como la inversión en el contexto político-económico que brinda nuestro país, constituyéndose en el gran mercado donde se encuentran tanto oferentes (ahorradores o inversionistas financieros) como demandantes (emisoras) de recursos monetarios o fondos.

Oferentes de capitales

Son personas físicas y morales que después de haber satisfecho sus necesidades propias, cuentan con un excedente de dinero para efectuar inversiones con el objeto de acrecentar su patrimonio. Las alternativas de inversión que tienen los oferentes de capital son:

- Mercado de Valores
- Instrumentos Bancarios
- Bienes Inmuebles
- Joyas, Obras de arte
- Mercado de Divisas
- Metales, etc

Demandantes de Capital

Son personas morales (empresas) que requieren de capital, el cual pueden obtener a través de distintas fuentes de financiamiento para desarrollar su actividad económica. Su fuente de financiamiento en el mercado de valores será a través de la emisión de instrumentos que componen los mercados de:

- Capitales
- Dinero
- Especiales

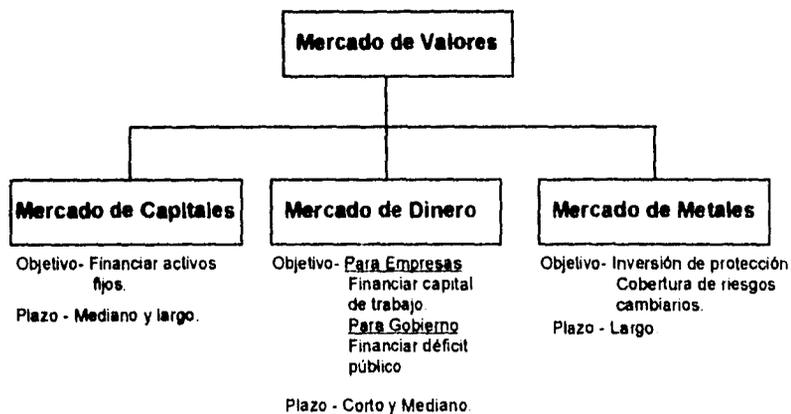


El mercado de valores es tan solo una alternativa de inversión dentro del Sistema Financiero Mexicano, se puede definir como el conjunto de leyes, reglamentos e instituciones tendientes a poner en contacto a la oferta (vendedores) y la demanda (compradores) para la compra-venta de valores a través de la Bolsa. Las instituciones que están involucradas de alguna forma con el mercado de valores son:

- Secretaría de Hacienda y Crédito Público
- Comisión Nacional de Valores
- Banco de México
- Bolsa Mexicana de Valores
- Asociación Mexicana de Casas de Bolsa A.C.
- Instituto para el Depósito de Valores

El mercado de valores esta dividido en el Mercado de Capitales, el Mercado de Dinero y el Mercado de Metales amonedados. El motivo de esta clasificación atiende a los diferentes destinos de los recursos captados, el plazo y el tipo de rendimiento que ofrecen.

CLASIFICACION DEL MERCADO DE VALORES



La clasificación de los recursos financieros por plazos de inversión o bien, por plazos de financiamiento dan origen a los conceptos del mercado de capitales y mercado de dinero principalmente, aunque actualmente podemos considerar otros mercados como el mercado de metales y los mercados especiales. A continuación se hablará en forma más detallada de los mercados de Capitales y de Dinero, ya que con parte de la información que estos mercados manejan, se trabaja dentro de la aplicación en estudio.



Mercado de Capitales

Es un conjunto de operaciones sobre emisiones de deuda de largo plazo y de mayor riesgo, así como operaciones sobre emisiones de renta variables. El mercado de capitales maneja instrumentos a largo plazo, los cuales su vigencia es superior a un año, sin embargo el poseedor puede venderlos en cualquier momento, su finalidad esta orientada a la formación de capital fijo. Para un análisis más detallado el mercado de capitales se puede subdividir en tres grupos:

- Instrumentos de deuda de largo plazo
- Acciones
- Productos Derivados

En cada una de estas subdivisiones, todos los días se realizan operaciones que al concretarse se convierten en hechos, todo esto se realiza en el piso de remates de la Bolsa Mexicana de Valores, efectuándose el mayor número de hechos con las acciones, siendo esta subdivisión la que más seguimiento se le da dentro del mercado de capitales.

Mercado de Dinero

Es el conjunto de operaciones sobre valores cuya vigencia es menor a un año, que se utilizan para financiamiento a corto plazo, y donde la emisoras pueden ser tanto el gobierno federal, como empresas privadas y bancos. El mercado de dinero opera generalmente con instrumentos de corto plazo, menores a un año, en ocasiones, con vencimientos inferiores a un mes.

Los inversionistas de este mercado buscan una rentabilidad fija para sus recursos con la expectativa de recuperarlos con prontitud. La clasificación que se utilizo dentro de las operaciones de mercado de dinero en la aplicación fue por tipo de emisor.

Gobierno	Bancario	Privado
Cetes	Aceptaciones Bancarias	Certificados de Depósito
Tesobonos	Bonos Prenda	Certificados de Participación
Ajustabonos	Bondis	Obligaciones
Bondes	BBDes	Pagare Mediano Plazo
	P.R.L.V.	Papel Comercial



Monitor de Mercados

Existen otras clasificaciones de los instrumentos que se manejan en el mercado de dinero tales como la forma en que se van a pagar o por el rendimiento que éstos otorgarán. El mayor volumen de hechos dentro del mercado de dinero que se realizan en el piso de remates es sobre los instrumentos que el gobierno emite. Para el desarrollo de esta aplicación no se consideraron monitoreo de operaciones realizadas en los mercados de metales y mercados especiales debido a que este tipo de información no la maneja la Bolsa Mexicana de Valores.



Análisis

Considerando que el desarrollo de esta aplicación, es tan solo uno de varios productos que se desarrollaron para la dirección de Análisis de la casa de bolsa, y siendo uno de esos productos el desarrollo de una base de datos integral que funga como servidor para las diferentes aplicaciones que se desarrollarán en un futuro, siendo una de ellas, la aplicación en estudio, se prestó especial atención en los requerimientos realizados por parte del usuario, ya que una de las claves para tener éxito en el desarrollo de un sistema dentro de la arquitectura cliente/servidor o en cualquier ambiente de desarrollo es antes de diseñar lo que se va a construir, es necesario recopilar, entender las necesidades y requerimientos de los usuarios para que la aplicación ha desarrollar satisfaga las expectativas que el usuario tiene.

Ya que los usuarios juzgan la calidad de un sistema en la medida en que este contemple sus requerimientos y además supere sus expectativas., a continuación se mencionan los puntos más importantes que se consideraron dentro de esta fase.

Propósito de la aplicación

Es importante determinar el objetivo desde el punto de vista del negocio que cualquier aplicación a desarrollar deberá alcanzar. En este caso el propósito de nuestra aplicación es monitorear las operaciones de compra-venta de los diferentes mercados (mercado de capitales, dinero, warrantes, etc) realizadas durante el día en el piso de remates de la Bolsa Mexicana de Valores, para un análisis posterior, así como la capacidad de realizar consultas históricas sobre un determinado mercado.

Forma de trabajo actual

Todos los usuarios requieren como fuente, la información de las operaciones que se realizan en el piso de remates de la Bolsa Mexicana de Valores, para ello requieren imprimir listados de esta información utilizando una terminal Tandem, que es propiedad de la Bolsa, o utilizando INFOSEL, generándose cuellos de botella, debido a que solo existe una sola terminal Tandem y un solo INFOSEL para todo el grupo de analistas.

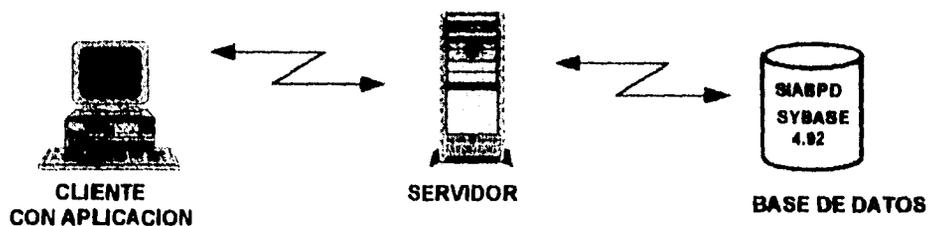
Usuarios que participaron y trabajarán en la aplicación

Conocer el perfil de usuario que estará participando en la definición de requerimientos y también los usuarios que finalmente utilizarán la aplicación. En este caso los usuarios son ejecutivos que tienen conocimientos en el ambiente Windows (herramienta con la que trabajan todos los días) y que parte de su trabajo es analizar la información de las operaciones de compra-venta realizadas en la Bolsa Mexicana de Valores y que todos los días es generada y enviada a través de la línea de comunicación que se tiene con la Bolsa.



Acceso a Datos

En las diferentes entrevistas que se tuvieron con los usuarios, se identificaron que tablas de la base de datos, la aplicación necesitaba acceder, así como las operaciones (insert, delete, update, retrieve) de sqls que se realizarán. En este caso, solo se harán operaciones de retrieves a las tablas por parte del usuario, ya que la alimentación de la mayoría de las tablas se realiza de manera automática (con excepción de las tablas de catalogos comunes).



La aplicación en estudio necesita acceder la base de datos denominada SIABPD (base de datos en SYBASE versión 4.92), misma que se encuentra en el servidor denominado DSQUERY. Mas adelante, en el diseño de pantallas se hablará a detalle de las tablas que a nivel ventana o consulta accesa esta aplicación.

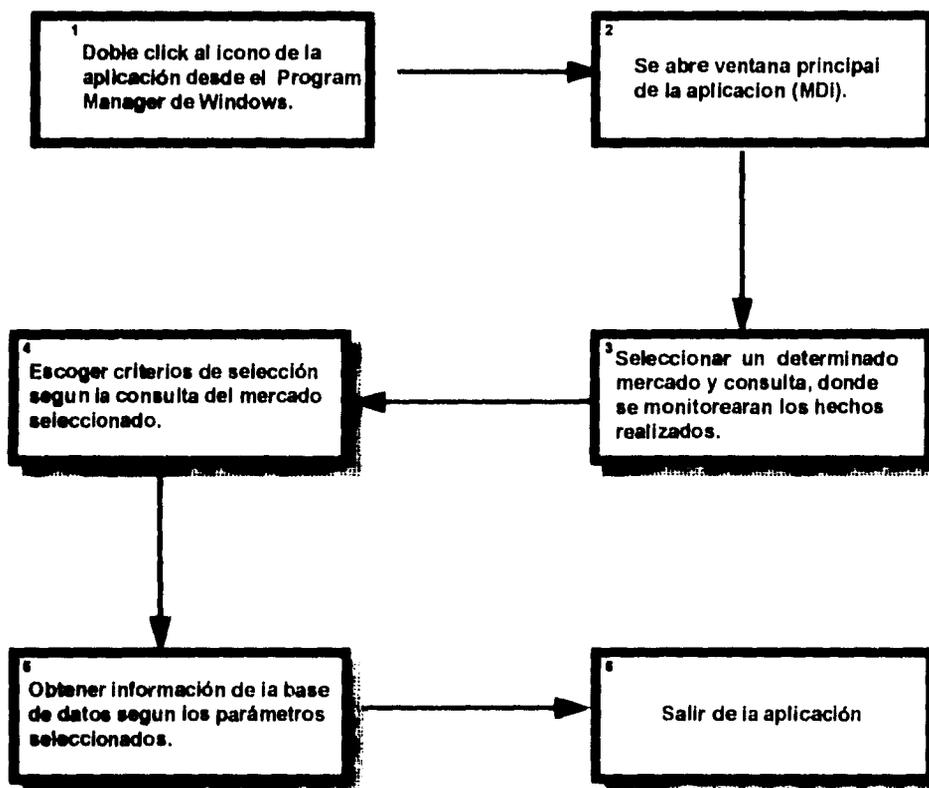
Interfase de Usuario

El usuario necesita conocer las operaciones de compra-venta sobre un determinado mercado, con cierta flexibilidad, además de que la aplicación fuera lo más parecida al ambiente windows, donde pudiera abrir una misma pantalla varias veces obteniendo diferente información dependiendo de las condiciones que se seleccionaron en cada una de ellas, teniendo la posibilidad de monitorear un mismo mercado (Renta Variable o de Dinero) al mismo tiempo pero con condiciones diferentes además de poderse mover de una pantalla a otra, identificándose de esta forma la necesidad de que la aplicación fuera MDI (Multiple Document Interface), siendo esta el estandar de las aplicaciones en Windows. Mas adelante, en el diseño de pantallas se hablará a detalle de la interfase de usuario que maneja esta aplicación.

Diagrama de flujo de la navegación

A continuación se muestra un diagrama de flujo general, donde se puede observar a manera de bloques, el flujo que en un determinado momento tendrá un usuario que quiera utilizar esta aplicación.

FLUJO GENERAL DE LA APLICACION

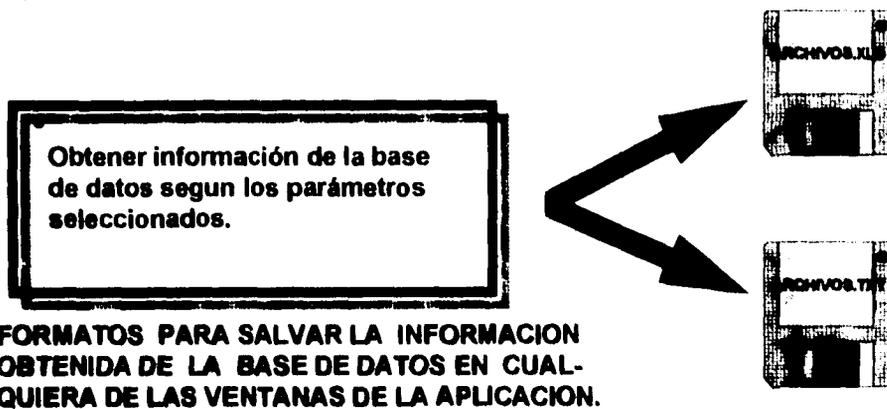


Dentro del bloque número dos, se selecciona el mercado y el tipo de ventana del mercado seleccionado a consultar, pudiéndose tener dentro del mismo espacio de trabajo, varias instancias abiertas de la misma ventana o ventanas de diferentes mercados. Los bloques que se encuentran sombreados no necesariamente siguen el flujo de la operación, ya que el usuario puede saltar de un mercado a otro y de una consulta o otra sin necesidad de tener que realizar acciones secuenciales. Mas adelante, en el diseño de pantallas se hablará a detalle de la interfase de usuario que maneja esta aplicación.



Salidas

Se identificaron que tipo de archivos van a ser generados con la información obtenida de una determinada ventana para un procesamiento posterior, considerándose en el menú de la aplicación una opción para salvar la información obtenida de la base de datos en los formatos TXT (formato tipo texto) y/o XLS (formato tipo excel).



Recordando la figura mostrada en el punto Diagrama de flujo de la navegación, es en el bloque numero cinco donde se tomará la decisión por parte del usuario, de bajo que formato se desea salvar la información obtenida de la consulta realizada. Adicionalmente dentro de este bloque, se puede mandar a imprimir los resultados obtenidos de la base de datos.

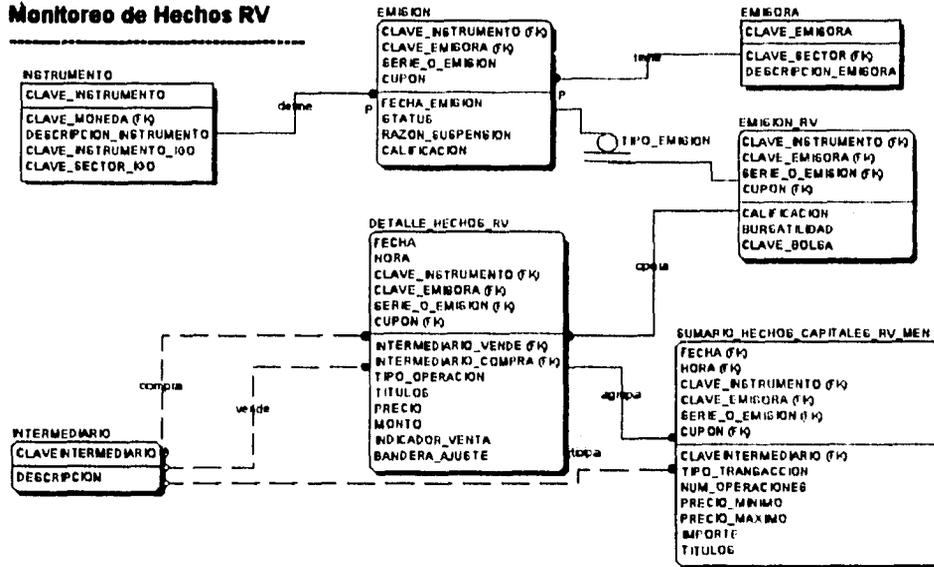
Modelo lógico de datos

Para el modelo lógico de datos se utilizó una herramienta lower CASE, totalmente compatible con el front-end utilizado, siendo esta, ERwin/ERX 2.1 FOR POWERBUILDER, herramienta con la que se puede realizar el modelado de datos. Debido a que con esta herramienta se va a realizar el modelaje de datos de otras aplicaciones incluyendo la aplicación en estudio, se trato de agrupar la información con respecto a cada aplicación, creandose diferentes areas sujeto (subject areas) por aplicación.

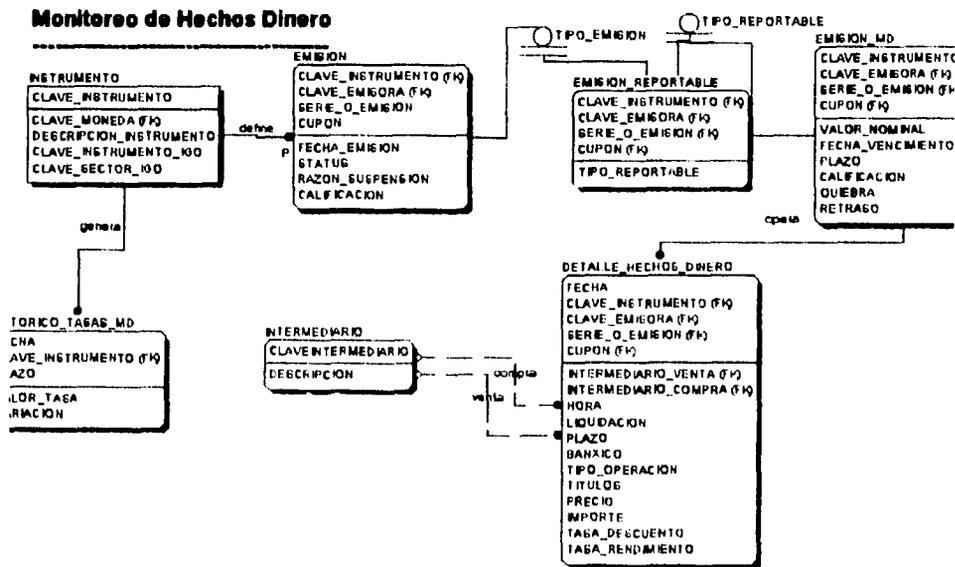
Para el caso de esta aplicación, se definieron areas sujetos por tipo de mercado, esto con el fin de que el usuario responsable de cada mercado conociera las información que su mercado maneja, presentandose a continuación como ejemplo las areas sujeto tanto del mercado de capitales de renta variable como el mercado de dinero.



Monitoreo de Hechos RV



Monitoreo de Hechos Dinero



Modelo Físico de datos

Una vez que se tiene un modelo lógico de datos capaz de reflejar las reglas del negocio que la aplicación demande, el siguiente paso es convertir las entidades, en tablas, y las relaciones entre entidades, en llaves primarias y secundarias de las tablas, obteniéndose de esta manera el modelo físico de datos de cualquier aplicación. Para esta aplicación, este paso se realizó de manera automática ya que la herramienta arriba mencionada, permite este tipo de acciones, evitando de esta manera errores de tipo humano al pasar del modelo lógico de datos al físico. A continuación se muestran como ejemplo scripts de creación en sybase de algunas de las tablas que se utilizan en esta aplicación.

```
CREATE TABLE dbo.INSTRUMENTO
    (CLAVE_INSTRUMENTO varchar(5) NOT NULL,
    DESCRIPCION_INSTRUMENTO varchar(50) NULL,
    CLAVE_MONEDA varchar(5) NOT NULL)

EXEC sp_primarykey 'dbo.INSTRUMENTO',
    'CLAVE_INSTRUMENTO';

CREATE UNIQUE CLUSTERED INDEX ind_INSTRUMENTO
    ON dbo.INSTRUMENTO
    (CLAVE_INSTRUMENTO)

CREATE TABLE dbo.EMISORA
    (CLAVE_EMISORA varchar(8) NOT NULL,
    DESCRIPCION_EMISORA varchar(90) NULL,
    CLAVE_SECTOR varchar(5) NOT NULL)

EXEC sp_primarykey 'dbo.EMISORA',
    'CLAVE_EMISORA';

CREATE UNIQUE INDEX ind_EMISORA
    ON dbo.EMISORA
    (CLAVE_EMISORA );

CREATE TABLE dbo.EMISION_RV
    (CLAVE_EMISORA varchar(8) NOT NULL,
    SERIE_O_EMISION varchar(8) NOT NULL,
```

CLAVE_INSTRUMENTO varchar(5) NOT NULL,
FECHA_COLOCACION datetime NULL,
REFERENCIA varchar(3) NULL,
FECHA_REFERENCIA datetime NULL,
MERCADO_PISO varchar(2) NULL,
CUPON_VIGENTE smallint NOT NULL,
VALOR_NOMINAL float NULL,
ACCIONES_EN_CIRCULACION float NOT NULL,
PRECIO float NULL,
CALIFICADOR_INTERNO varchar(6) NULL,
CALIFICADOR_EXTERNO varchar(6) NULL,
INDICADOR_BURSATIL float NULL,
INDICADOR_INDICE varchar(2) NULL,
PERSONALIDAD varchar(3) NULL,
RESIDENCIA varchar(3) NULL,
RESTRICCIONES varchar(50) NULL,
VALORES_X_TITULO float NULL,
VALORES_X_TITULO_AJU float NULL)

```
EXEC sp_primarykey 'dbo.EMISION_RV',  
    'CLAVE_EMISORA',  
    'SERIE_O_EMISION',  
    'CLAVE_INSTRUMENTO';
```

```
EXEC sp_foreignkey 'dbo.EMISION_RV',  
    'dbo.EMISION',  
    'CLAVE_EMISORA',  
    'SERIE_O_EMISION',  
    'CLAVE_INSTRUMENTO';
```

```
CREATE UNIQUE INDEX ind_EMISION_RV  
ON dbo.EMISION_RV  
(CLAVE_EMISORA ,  
SERIE_O_EMISION ,  
CLAVE_INSTRUMENTO );
```



```
CREATE TABLE dbo.DETALLE_HECHOS_CAPITALES_RV
```

```
(CLAVE_EMISORA varchar(8) NOT NULL,  
FECHA smalldatetime NOT NULL,  
FOLIO int NULL,  
HORA int NULL,  
CLAVE_INSTRUMENTO varchar(5) NOT NULL,  
SERIE_O_EMISION varchar(8) NOT NULL,  
TITULOS float NOT NULL,  
PRECIO float NOT NULL,  
TIPO_OPERACION varchar(2) NULL,  
INTERMEDIARIO_COMPRA varchar(8) NOT NULL,  
INTERMEDIARIO_VENDE varchar(8) NOT NULL,  
CUPON smallint NOT NULL,  
MONTO float NULL,  
INDICADOR_VENTA varchar(3) NULL,  
BANDERA_AJUSTE bit NOT NULL,  
TIPO_MERCADO varchar(2) NULL);
```

```
CREATE UNIQUE CLUSTERED INDEX ind_DETALLE_HECHOS_CAPI_RV
```

```
ON dbo.DETALLE_HECHOS_CAPITALES_RV  
(CLAVE_EMISORA ,  
SERIE_O_EMISION ,  
CLAVE_INSTRUMENTO ,  
FOLIO ,  
FECHA );
```

```
CREATE INDEX ind_DET_HEC_CAPIRV_FECHA
```

```
ON dbo.DETALLE_HECHOS_CAPITALES_RV  
(FECHA ,  
CLAVE_EMISORA ,  
SERIE_O_EMISION );
```

```
CREATE INDEX ind_DET_HEC_CAPIRV_COMPRA
```

```
ON dbo.DETALLE_HECHOS_CAPITALES_RV  
(INTERMEDIARIO_COMPRA );
```



```
CREATE INDEX ind_DET_HEC_CAPIRV_VENDE
ON dbo.DETALLE_HECHOS_CAPITALES_RV
(INTERMEDIARIO_VENDE );
```

```
CREATE TABLE dbo.SUMARIO_HECHOS_CAPITALES_RV
(FECHA smalldatetime NOT NULL,
CLAVE_EMISORA varchar(8) NOT NULL,
CLAVE_INSTRUMENTO varchar(5) NOT NULL,
SERIE_O_EMISION varchar(8) NOT NULL,
CLAVE_INTERMEDIARIO varchar(8) NOT NULL,
TIPO_TRANSACCION varchar(3) NULL,
NUM_OPERACIONES float NULL,
PRECIO_MINIMO float NULL,
PRECIO_MAXIMO float NULL,
IMPORTE float NULL,
TITULOS float NOT NULL);
```

```
CREATE INDEX ind_SUMARIO_HECHOS_CAPI_RV
ON dbo.SUMARIO_HECHOS_CAPITALES_RV
(CLAVE_INSTRUMENTO,
CLAVE_EMISORA,
SERIE_O_EMISION,
FECHA );
```

```
CREATE TABLE dbo.INTERMEDIARIO
(CLAVE_INTERMEDIARIO varchar(8) NOT NULL,
DESCRIPCION varchar(50) NULL);
```

```
EXEC sp_primarykey 'dbo.INTERMEDIARIO',
'CLAVE_INTERMEDIARIO';
```

```
CREATE UNIQUE INDEX ind_INTERMEDIARIO
ON dbo.INTERMEDIARIO
(CLAVE_INTERMEDIARIO );
```



Monitor de Mercados

Tanto el modelo lógico y físico de datos son procesos evolutivos en todo momento y en la medida que se van teniendo entrevistas con el usuario, muestra de prototipo, etc, donde es posible detectar reglas de negocio no consideradas, o simplemente abrir el alcance de la aplicación, teniéndose que ajustar el modelo lógico y por lo tanto el físico. Siempre se debe tener un modelo lógico y físico de datos en línea, esto es, que cuando se ajuste el modelo lógico de datos se actualice el físico y al revés, y con la ayuda de esta herramienta es posible alcanzar ese objetivo.

Nota. - Es importante recalcar que este estudio no considera hablar profundamente del diseño lógico y físico de datos siendo los responsables personas diferentes al responsable del desarrollo de la aplicación, pero no por ello implicó la no participación en esta fase del proyecto, mencionándose tan solo los productos finales.



Diseño Conceptual de la Aplicación

Configuración de ambiente

Para el desarrollo de esta aplicación la configuración del ambiente se dividió en los siguientes componentes:

- Servidores

Se utilizaron como servidores una HP735 y una SUN con las siguientes características físicas:

Estación de Trabajo HP735:

- Sistema Operativo HP-UX versión 9.01.
- Memoria RAM 128 MB.
- Disco interno de 1GB
- Configuración en área de swap de 340MB.
- Dirección en red ETHERNET 150.108.101.201.
- Tarjeta Ethernet integrada.

Estación de Trabajo SUN:

- Sistema Operativo Solaris Ver. 2.3.
- Memoria RAM 16 MB.
- Disco interno de 1 GB.
- Dirección en red ETHERNET 150.108.101.206.210.
- Tarjeta Ethernet integrada.

El software operativo con el que se trabajó fue:

Sistema Operativo

- UNIX (HP-UX) v 9.01
- Solaris v 2.3

Utilerias UNIX:

- Data WorkBench Mantenimiento a los Datos.
- vi Editor de programas.
- Compilador C. Programas de Inyección.



Monitor de Mercados

- DBMS

Se utilizó como manejador de base de datos a Sybase versión 4.9.2

- Hardware y software de red

- Cliente de desarrollo

Se utilizó una PC COMPAQ con las siguientes características físicas

- * Procesador 486/33 megahertz.
- * Sistema Operativo DOS versión 5.0 o mayor.
- * Memoria RAM 8 MB.
- * Disco duro de 120 MB.
- * Tarjeta de red integrada.
- * Monitor color SuperVGA.
- * Mouse.

El software operativo con el que se trabajó fué :

Sistema Operativo

- * DOS v 5.0 o mayor
- * Windows 3.1

Aplicaciones Windows

- * PowerBuilder versión 3
- * Word versión 2 (Documentación)
- * Powerpoint versión 2 (Documentación)

Estandares y Convenios

Debido a que uno de los objetivos es construir una aplicación con funcionalidad y calidad, fué necesario definir una serie de estandares y convenios en los siguientes puntos

- ◆ Interfase de usuario GUI's
- ◆ Nombres
- ◆ Documentación

A continuación se habla más a detalle de cada uno de estos puntos



Interfase de usuario GUI's

Debido a que el ambiente de trabajo del usuario final es Windows, se trato de aprovechar esta situación, para que el desarrollo de las GUI's fuera similar a éste. PowerBuilder proporciona una variedad de herramientas y opciones que se pueden utilizar para construir GUI's bajo un ambiente de Windows.

El estilo de GUI's que se utilizó fue MDI debido a su flexibilidad, ya que se pueden tener abiertas al mismo tiempo varias ventanas e intercambiarlas entre si.

Los convenios que se utilizaron para el diseño de los objetos ventanas y los controles que esta puede tener se describen en la siguiente tabla.

TEMA	CONVENIO
Tipos de controles	Para que sea una interface intuitiva , utilizar el control adecuado para cada feature que se quiera implementar en la ventana.
Disponibilidad de controles	Deshabilitar un control cuando este no esta disponible para el usuario.
Número de controles	Limitar el número de controles que se colocan en la ventana para no sobrecargar al usuario.Utilizar más ventanas de ser necesario
Longitud de las listas de control	Limitar el número de campos en ListBox o DropDownListBox a desplegar.
Colores	Utilizar colores de default. El color de fondo de las ventanas será gris por petición del usuario final
Fonts	Limitar el número de fonts de uno a dos.Para el texto se utilizó el font Arial, tamaño 10, color negro.



Los convenios que se utilizaron para el diseño del objeto menú se describen en la siguiente tabla.

TEMA	CONVENIO
Disponibilidad de un campo del menú	Deshabilitar un campo del menú si este no está disponible para el usuario.
Longitud en el nombre de un campo del menú	Tratar que los campos del menú sean de una o dos palabras. (Nunca más de cuatro)
Significado de los campos del menú	Tratar de que el nombre del campo del menú sea descriptivo de lo que éste realiza.
Ayuda	Deberá existir un campo de menú de ayuda, siendo este el último campo del menú.
MDI micro ayuda	Brindar microayuda en forma de texto.
Teclas de rápido acceso	Proporcionar teclas de rápido acceso sobre cualquier campo del menú (por ejemplo <u>F</u> ile, <u>E</u> dit).

El seguimiento de estándares y convenios para desarrollar cualquier aplicación permite establecer consistencia entre varias aplicaciones, así como facilita el reutilizar componentes de otras aplicaciones, evitando de esta manera redundancia en el diseño y eliminando tiempo de desarrollo para futuras aplicaciones.

Nombres

Al utilizar PowerBuilder se utilizan diferentes componentes, a los que hay que asignarles algún nombre para poderlos identificar, por lo tanto es importante mencionar los objetos de PowerBuilder con los que se trabajó para el desarrollo de esta aplicación (ver tabla de abajo) El nombre de cualquiera de estos objetos puede ser hasta de 40 caracteres, utilizándose un prefijo dependiendo del tipo de objeto.

TIPO DE OBJETO	PREFIJO	EJEMPLO
Ventana (Window)	w	w_ab0201
Menu	m	m_principal
Usuario Objeto (User Object)	u	uo_fecha
Ventana de datos (Data Window)	dw	dw_hechos_rv
Función	f	f_salvar
Aplicación	ninguno



Dentro del objeto de tipo ventana se utilizaron diferentes controles, y que al igual que los objetos arriba mencionados, se utilizó un prefijo dependiendo del tipo de objeto.

TIPO DE CONTROL	PREFIJO	EJEMPLO
Check Box	cbx	cbx_banxico
DataWindow	dw	dw_emisoras
DropDownListBox	ddlb	ddlb_instrumentos
User Object	uo	uo_fecha
StaticText	st	st_titulo
Radio Button	rb	rb_tipo_mercado

Este tipo de convenios para los nombres de los diferentes objetos que se utilizaron, es importante definirlos en esta etapa dentro del desarrollo de la aplicación, ya que se ahorra tiempo tanto en el desarrollo como en el soporte a la producción, teniendo un mayor control de los objetos involucrados.

Documentación

Una buena documentación es esencial para el desarrollo de cualquier aplicación. La documentación que se generó para esta aplicación fue un tanto interna, la cual se generó en el transcurso del desarrollo (documentación que puede servir para posibles cambios en la aplicación), como externa útil para el usuario final.

Para la documentación interna, se tecló una descripción asociada a cada objeto (ventana, datawindow, user object, etc) que se utilizó en la aplicación, tratando de que el comentario fuera lo más descriptivo posible con el fin de que al leerlo se pueda saber que es lo que el objeto realiza.

```

- [ ] m_ab0203      1/3/95 14:29:00 (55245) Consulta de Hechos de Renta Fija
- [ ] m_ab0204      1/3/95 14:29:17 (43220) Consulta de Hechos de Derivados
- [ ] m_ab0205      1/3/95 14:30:13 (13324) Consulta Movimiento Neto por Intermediario (Global)

```

Adicionalmente en cada script de cualquier evento, se tecló en la parte de arriba un comentario explicando su propósito.

```

//*****
/*Window:          w_hechos_rv
/*Evento:          Open
/*Descripción:     Se inicializar los objetos fecha_del, fecha_al y se carga la lista de
/*                intermediarios
/*Librerías dinámicas: siabl7.pbl
/*Lógica especial: Se llama a la función f_lista_intermediarios
/*

```



Monitor de Mercados

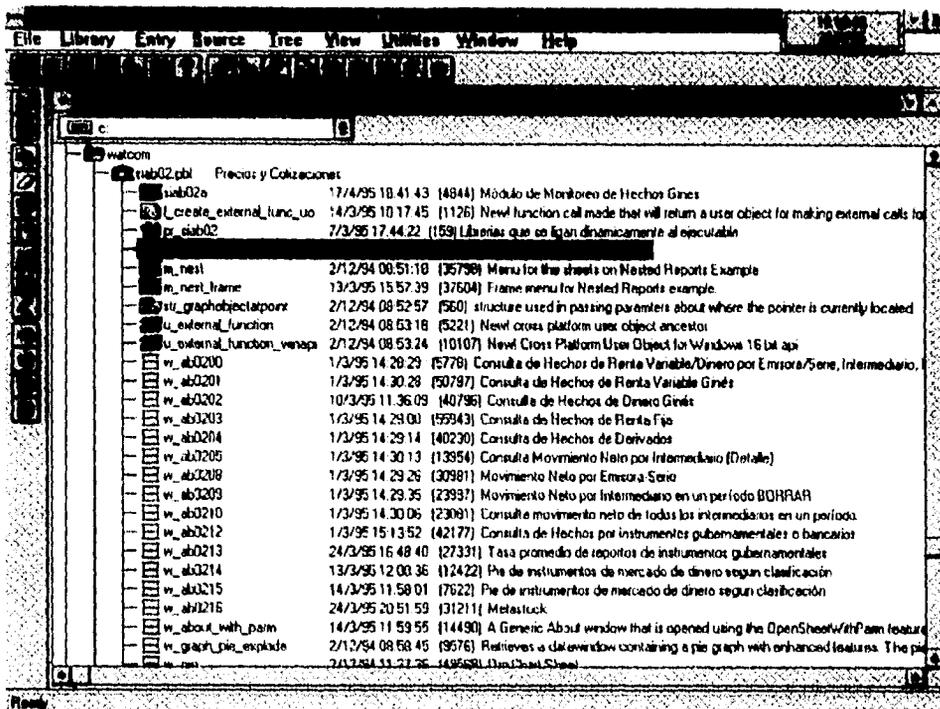
```
//*****  
//*   Dia           Programador  
//*   -----  
//*   01-Mayo-95   Ginés Rivero  
//*****
```

Como documentación externa el usuario final utiliza la ayuda en línea cuando este se encuentra trabajando con la aplicación, con el fin de resolver algún tipo de duda referente a lo que hace la aplicación o a lo que se puede realizar dentro de una determinada ventana.

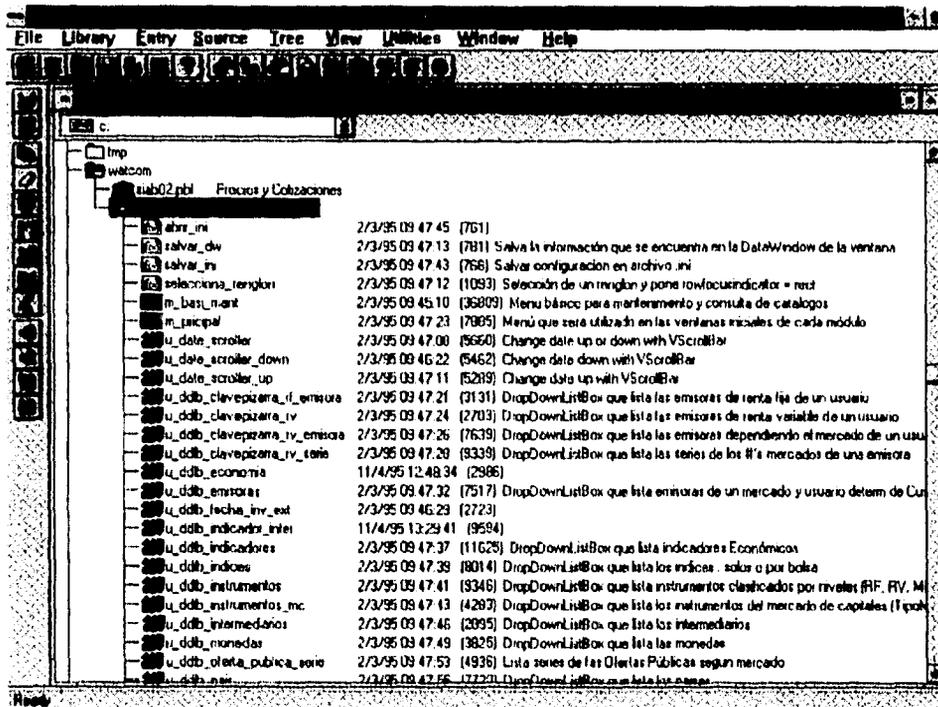


Organización de objetos

La colección de objetos propios de la aplicación que se fueron creando durante el desarrollo, fueron almacenados en el archivo siab02.pbl (archivo fuente), este archivo dentro de PowerBuilder es visto como un contenedor de objetos denominado Librería, en donde cada tipo de objeto dentro de la librería es representado por una figura diferente (ver figura a continuación).



Aprovechando la facilidad que da PowerBuilder para manejar el encapsulamiento de características, el comportamiento (atributos, eventos, funciones), herencia, y polimorfismo de los objetos que se crean (bases de la teoría orientada a objetos), se utilizó un **Application Framework** (librería siab17.pbl), en donde se agrupan objetos con el fin de reutilizarlos en otras aplicaciones o dentro de la misma aplicación (ver figura a continuación).



Nota. No todos los objetos que aparecen en esta librería fueron utilizados para el desarrollo de esta aplicación

Al definirse estos objetos clase con ciertas características y con un determinado comportamiento (atributos, eventos, funciones), se pueden crear objetos heredados de estos, adquiriendo el nuevo objeto todas las características y comportamiento del objeto clase de manera automática.

Existe gran variedad de productos del tipo de application framework en el mercado, disponibles que se pueden utilizar dentro de PowerBuilder, para este caso de estudio parte de la librería 17 (siab17.pbl) fue desarrollada durante la generación de esta aplicación.



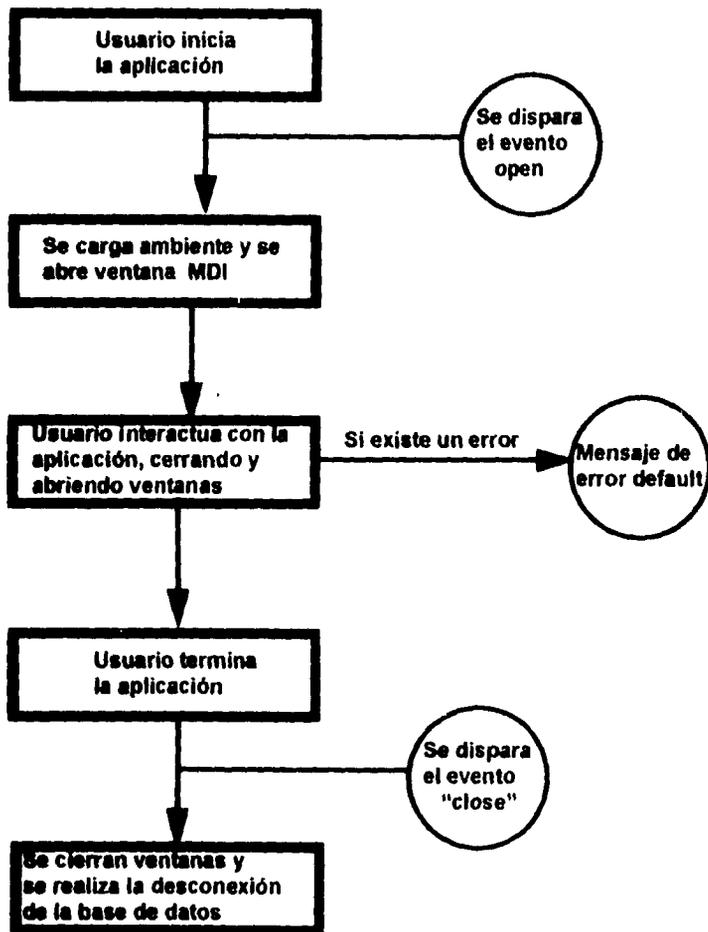
Logística de la Aplicación

Hasta este momento se han considerado puntos importantes dentro de esta aplicación tales como: la definición del hardware y software, definición de estándares y convenios, el manejo de las librerías y bases de datos, faltando por definir un objeto importantísimo para el desarrollo de cualquier aplicación dentro de PowerBuilder. Este objeto recibe el nombre de **application object**, el cual nos permite definir la logística que tendrá esta aplicación, definiéndose en él los siguientes puntos:

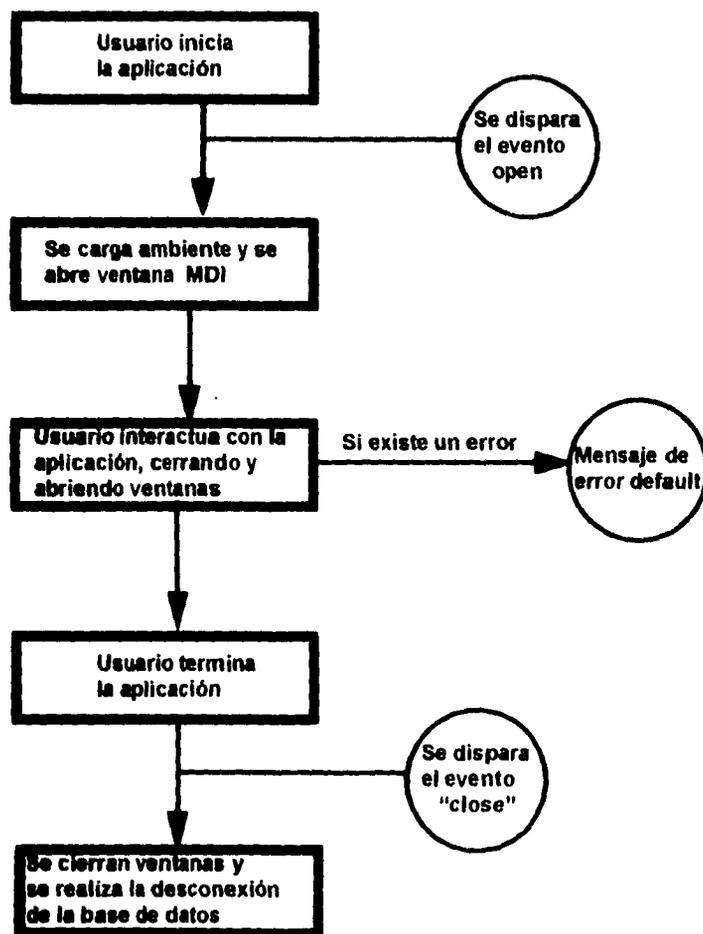
- La lista de librerías que maneja la aplicación
PowerBuilder utiliza esta lista para encontrar objetos durante el desarrollo y ejecución de la aplicación, realizando la búsqueda en las librerías en el orden definido en la lista.
- El icono que el sistema operativo desplegará para representar la forma ejecutable de la aplicación.
- Los procesos que se van a realizar cuando :
 - ⇒ Un usuario inicie(abra) la aplicación.
El objeto **application object** cuenta con el evento "open", el cual se dispara cuando el usuario inicia la aplicación (dar doble click al icono de la aplicación que se encuentra en el grupo del SIAB dentro del program manager de windows). Se escribió un script para este evento en donde, se inicializan las variables globales, se cargan las variables de ambiente (variables definidas en el archivo siab02.ini , se realiza la conexión a la base de datos y se abre la ventana MDI principal
 - ⇒ Se presenta un error de sistema cuando un usuario esta en la aplicación
Se maneja para esta aplicación el mensaje de error default que envia PowerBuilder, el cual contiene el número de error y texto adecuado, clarificando de esta manera el error.
 - ⇒ Un usuario termine (cierre) la aplicación.
El objeto **application object** cuenta con otro evento "close", el cual se dispara cuando el usuario termina la aplicación. El proceso que se dispara en el script de este evento antes de cerrar la aplicación, es la desconexión de la base de datos.
- Las variables de tipo global que se utilizaron
- El tipo de font que se utilizo por default en la aplicación para encabezados, textos, etc



A continuación se presenta un diagrama de flujo general de la lógica de la aplicación en estudio, en donde se identifican tres eventos dentro del application object, eventos que controlarán a nuestra aplicación.



A continuación se presenta un diagrama de flujo general de la lógica de la aplicación en estudio, en donde se identifican tres eventos dentro del application object, eventos que controlarán a nuestra aplicación.





Monitor de Mercados

Nota : La carga del ambiente se lleva a cabo leyendo el archivo `siab02.ini`, el cual tiene las siguientes variables definidas dentro del grupo `SQLCA`:

```
[SQLCA]
dbms=ODBC
database=c:\watcom\precios.db
userid=dba
dbpass=sql
logid=
logpass=
servername=
DbParm=ConnectString='DSN=PRECIOS;UID=dba;PWD=sql'
```

El papel que desempeña el **application object** en tiempo de ejecución es importantísimo, ya que como podemos observar en el diagrama de flujo anterior, este objeto es el que controla la aplicación en un nivel macro.



Desarrollo de la Aplicación

Una vez que se ha trabajado en los puntos anteriores (roles de trabajo, definición de estándares, logística de la aplicación, etc.), podemos empezar a diseñar objetos y codificar dentro de ellos *scripts*, desarrollando de esta manera las características propias de esta aplicación.

Para el desarrollo de esta aplicación se tuvo especial atención en la interfase de usuario (GUI), tratando de que esta fuera fácil de entender y de operar, aprovechando la metodología RAD. Se presentó un prototipo de lo que sería la aplicación, dimensionando las expectativas del usuario, así como para que conociera como trabajaría con la aplicación. De tal forma que se detectarían de forma inmediata errores o malos entendidos, pudiéndose realizar los cambios necesarios y obteniéndose las siguientes ventajas en el desarrollo:

- Construcción acelerada de la aplicación.
- Realización de pruebas de manera inmediata.
- Responder a comentarios y nuevos requerimientos conforme se van generando.
- Ganancia de tiempo en cuanto al desarrollo de la aplicación

El prototipo creció y evolucionó, si consideramos que inició como un esqueleto básico, y conforme el proyecto fue avanzando se le fueron agregando mas características, para que nuevamente fuera sometido a pruebas, asegurándonos de esta manera que la aplicación no estuviera lejos de lo que la gente quería. La estrategia que se siguió para llevar esta aplicación a un punto en que el usuario pudiera utilizarla como prototipo fue:

- Creando el mayor número de ventanas
- Pintando las características visuales más importantes en las ventanas.
- Creando el código suficiente para poder desplegar las ventanas y el usuario pudiera navegar entre ellas

Todo esto con el fin de que el usuario se pueda dar cuenta a tiempo de lo que su aplicación será, teniendo como resultado una importante retroalimentación en un momento oportuno, ayudando a llevar al proyecto hacia una correcta dirección.

También es importante considerar que la herramienta que se utilizó para el desarrollo de la aplicación (PowerBuilder), esta diseñada para soportar la metodología RAD, permitiendo construir, diseñar y realizar pruebas, identificándose diferentes actividades base dentro de esta etapa:



De tal forma que estando en el desarrollo de algún componente (ventana, datawindow, control de una ventana, función, aplicación, etc.), se puede realizar cualquiera de las actividades base arriba mencionadas, sin importar el nivel del componente. A continuación se hablará a detalle de estas tres actividades.

Construcción

Para la construcción de objetos, fue necesario identificar los objetos comunes que la aplicación iba a tener, así como aprovechar los conceptos de herencia y polimorfismo que PowerBuilder maneja y considerando las características que tiene esta aplicación, se crearon objetos base acelerando el tiempo de desarrollo y facilitando su mantenimiento. A continuación se hablará de algunos de los objetos base que se utilizaron en el desarrollo de esta aplicación.

w_sheet_dw1

Objeto de tipo ventana, que es tomada como base (ancestro) por las diferentes ventanas que se utilizan en esta aplicación y tiene como objetivo responder a las diferentes acciones (obtener, primero, siguiente, anterior, última) que el usuario puede tomar en algún momento dado. Para responder a estas acciones se definieron dentro de este objeto, eventos de usuario (user events), siendo estos los siguientes:

- ue_obtener
- ue_primer
- ue_sigueme
- ue_anterior
- ue_ultimo
- ue_imprimi



Monitor de Mercados

- ue_salir

Cada uno de estos eventos, son disparados cuando el usuario selecciona una determinada acción desde el menú asociado a la ventana MDI o utilizando la figura (toolbar) asociada a la acción del menú (en el diseño de la ventana W_MDI_HECHOS se hablará a detalle de estas acciones).

Dentro de este objeto existe un control datawindow que se llenará o no de datos, dependiendo de las condiciones seleccionadas de cada ventana, siendo en este control y concretamente en el evento **retrievestart** donde se manipula la estructura sql asociada al objeto según las condiciones seleccionadas, cambiando dinámicamente el sql a ejecutarse y haciendo más flexible la consulta realizada.

Solo se maneja un solo nivel de herencia con este objeto (siendo este el objeto padre y todas las ventanas de la aplicación las ventanas hijas), esto con el fin de buscar un mejor desempeño, ya que entre más niveles de herencia se tengan entre los objetos, es más pesado cargarse en memoria la aplicación.

u_date_scroller_down

Objeto de tipo user object, que permite decrementar o incrementar una fecha con la ayuda del mouse o teclear una fecha determinada, validándose que lo que se teclea es una fecha válida.

uo_emisora_serie_rv

Objeto de tipo user object, que es llamado en el evento **open** de las ventanas que pertenezcan al mercado de renta variable, su objetivo es que el usuario seleccione una determinada emisora-serie de la tabla de EMISION_RV.

uo_emisora_serie_derivados

Objeto de tipo user object, que es llamado en el evento **open** de las ventanas que pertenezcan al mercado de derivados, su objetivo es que el usuario seleccione una determinada emisora-serie de la tabla de EMISION_DERIVADOS.

uo_emisora_serie_rf

Objeto de tipo user object, que es llamado en el evento **open** de las ventanas que pertenezcan al mercado de renta fija, su objetivo es que el usuario seleccione una determinada emisora-serie de la tabla de EMISION_RF.

uo_lista_intermediarios

Monitor de Mercados

Objeto de tipo user object , que es llamado en el evento **open** de la mayoría de las ventanas de esta aplicación y que su objetivo es mostrar una lista de intermediarios (casas de bolsa) para su selección.

Cada uno de estos objetos, fue utilizado en más de una ventana dentro de la aplicación, y estos fueron evolucionando en la medida que se tuvieron entrevistas con los diferentes usuarios responsables, definiéndose en algunos de ellos nuevas características, modificándose y en forma automática reflejándose los cambios en las ventanas donde se utilizan. Estos objetos se encuentran agrupados físicamente en la librería **siab17.pbl** (application framework) mencionada en capítulos anteriores.



Diseño de pantallas

Para el diseño de las ventanas se designaron por parte del usuario representantes o responsables de la información de cada uno de los mercados generadores de hechos, esto con el fin de tener sesiones de trabajo productivas con usuarios que todos los días manipulan la información que la bolsa envía de un determinado mercado.

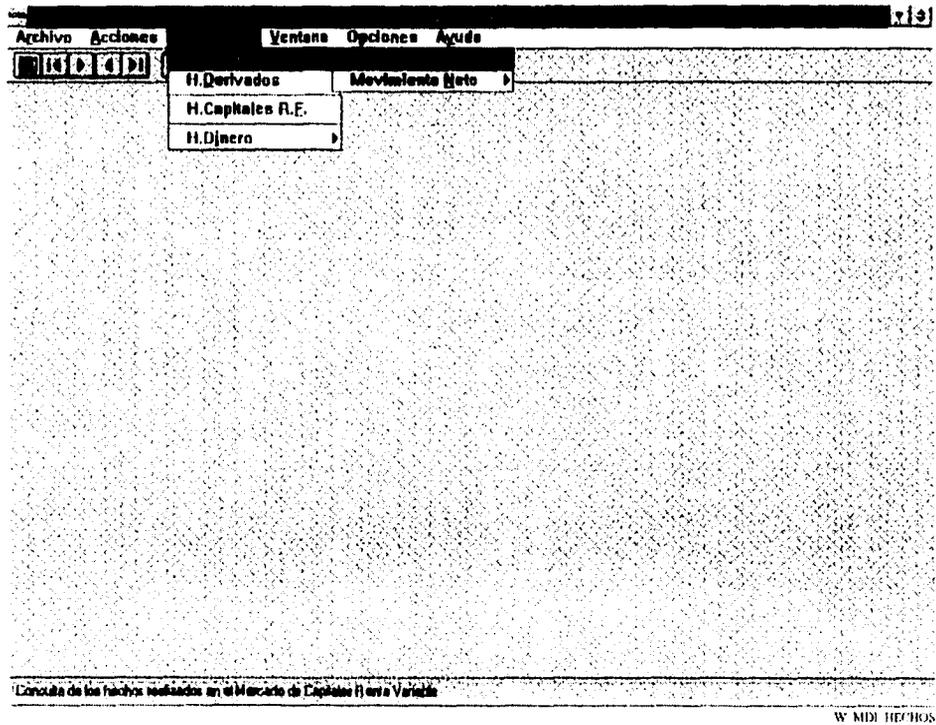
Se asignaron cuatro responsables por cada área de información, reflejándose en el diseño del menú, ya que éste cuenta con cuatro tipos de mercado de donde se pueden realizar consultas de los hechos

- Hechos Capitales R.V.
- Hechos Derivados
- Hechos Capitales R.F.
- Hechos Dinero

Conjuntamente con el responsable se trabajó para el diseño de la(s) consulta(s) de cada mercado con el fin de que las pantallas diseñadas pudieran ser utilizadas por ellos mismos o por gente que en ocasiones les pedía información sobre determinado mercado.

Pensando que los responsables se convierten en agentes prestadores de servicio hacia otras direcciones dentro de la Casa de Bolsa y que esta aplicación sería distribuida fuera, se trataron de diseñar consultas con las condiciones suficientes para que puedan ser utilizadas por gente que regularmente se ponía en contacto con el responsable para pedirle información sobre un determinado mercado, convirtiéndose la aplicación en el nuevo prestador de servicio, haciendo más ágil la comunicación entre el prestador de servicio y el usuario final.

A continuación se presentan las ventanas que la aplicación en estudio maneja en tiempo de ejecución, cada una de ellas con una explicación de su funcionamiento, la información que manejan y una lista sobre las tablas que accesan.



Funcionamiento

Ventana de tipo **MDI** que es llamada desde el evento `open` del `Aplicación Object`, dentro de la cual se abrirán todas las consultas (ventanas) que la aplicación maneja, para ello se necesita que los usuarios seleccionen la consulta que se desea realizar, teniendo que utilizar un objeto de tipo menú (`m_hechos`), el cual se encuentra asociado a esta ventana. Este objeto controla las ventanas que el usuario vaya seleccionando.

Dentro del `menúitem "Archivo"`, se consideraron dos funciones para poder manipular posteriormente la información obtenida de las consultas de la aplicación, siendo estas las siguientes:

- Imprimir  y salvar resultados



Monitor de Mercados

Solo aparecerán habilitadas cuando en la consulta realizada se obtenga información de la base de datos.

- Salir

Esta función cerrará la aplicación sin importar el número de ventanas que se encuentren abiertas al momento de su selección.

Dentro del menúitem "Acciones", se consideraron las posibles acciones que el usuario podría realizar en determinado momento con la información obtenida de la base de datos. A continuación se explica brevemente cada una de ellas.

- Obtener Información

Esta opción aparece habilitada en todo momento de la aplicación y se utiliza cuando en una ventana, (ventana dentro de la MDI) se desea obtener cierta información dependiendo de los parámetros seleccionados según sea el caso.

- Primer registro

Una vez que se ha traído la información deseada y se desea posicionarse en el primer renglón de los registros obtenidos.

- Registro Siguiente

Una vez que se ha traído la información deseada, el cursor se encuentra en uno de los renglones de la información obtenida y se desea seleccionar el siguiente renglón hacia abajo.

- Registro Anterior

Una vez que se ha traído la información deseada, el cursor se encuentra en uno de los renglones de la información obtenida y se desea seleccionar el siguiente renglón hacia arriba del seleccionado previamente.

- Último registro

Una vez que se ha traído la información deseada, y se desea seleccionar el último registro de nuestra consulta.



Monitor de Mercados

- Salir de la ventana

Se cierra la ventana que este activa (en la que se este trabajando) haciendo activa a la ventana anterior a ésta. de no haber ventana anterior se saldrá de la aplicación.

Dentro del menuitem "Consultas", se clasificaron 2 grupos de consultas, uno sobre información referente al mercado de capitales y el otro sobre información del mercado de dinero. En este punto solo se mencionarán ya que más adelante se explicará cada una de ellas. Para mercado de capitales tenemos la siguiente clasificación: de ventanas

- H. Capitales R.V.
 - Hechos R.V +
 - Movimiento Neto
 - Por Emisora +
 - Por Intermediario +
- H. Derivados. +
- H. Renta Fija +

Para Mercado de Dinero tenemos la siguiente clasificación de ventanas:

- H. Dinero
 - Por Instrumento +
 - Por Emisión +

Nota : Solo las que tienen la marca + son ventanas.

Dentro del menuitem "Ventanas", se consideraron las posibles opciones que el usuario podría realizar sobre las ventanas abiertas dentro de la MDI (igual que el File Manager de Windows). A continuación se explica brevemente cada una de ellas.

- Cascada

Opción que acomoda las ventanas abiertas dentro de la MDI en forma de cascada.

- Mosaico

Opción que acomoda las ventanas abiertas dentro de la MDI en forma contigua una de otra. llenándose el espacio del cliente del MDI.



Monitor de Mercados

Opción que acomoda las ventanas abiertas dentro de la MDI en forma contigua una de otra. llenándose el espacio del cliente del MDI.

- Llenar

Opción que toma la ventana activa, llenando el espacio del cliente del MDI

Dentro del menú "Ayuda", se consideraron las posibles opciones que el usuario podría seleccionar en busca de ayuda. Estas opciones son:

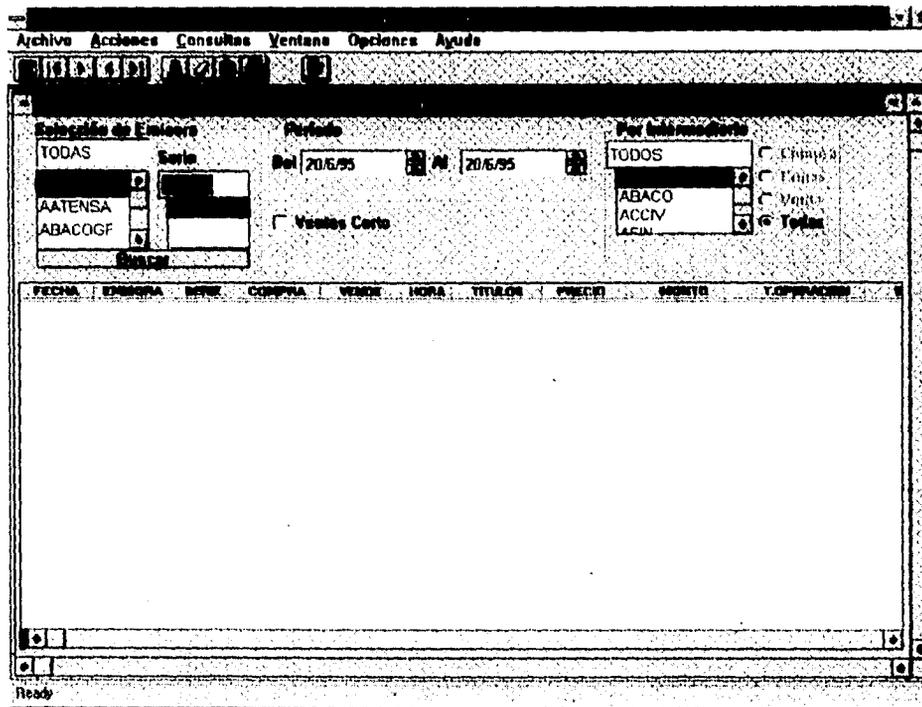
- Índice

Muestra los diferentes temas en que se encuentra dividida la ayuda de esta aplicación.

- Buscar Tema

Permite buscar un tema específico de la aplicación para su ayuda y seleccionarlo.

Ventana W_HECHOS_RV



W_HECHOS_RV

Funcionamiento

Ventana que es utilizada para consultar todos los hechos realizados sobre emisiones de renta variable en un periodo determinado (por default día de hoy), según ciertas condiciones que el usuario seleccione. Lo que se busca en esta ventana, es monitorear todos los hechos de todas las emisoras que se han realizado en el día o sobre alguna emisora en particular, o las operaciones que esta efectuando un determinado intermediario sobre alguna emisora, etc (La información obtenida dependerá del criterio de selección que se haya escogido). El tiempo de retraso que se tiene con respecto a la hora en que se realizó el hecho es de minuto y medio, que es el tiempo que tarda en llegar de la Bolsa Mexicana de Valores a nuestra base de datos.

También es posible realizar consultas sobre hechos en días pasados considerando los mismos criterios de selección, pero no es el objetivo primordial de esta pantalla.



Monitor de Mercados

Se tiene en línea hasta tres meses de hechos de renta variable y el promedio de hechos que se pueden recibir en un día es de 2000.

Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Fecha
- Hora
- Emisora
- Serie
- Comprador
- Vendedor
- Títulos
- Precio
- Monto
- Tipo de Operación
- Indicador de venta en corto

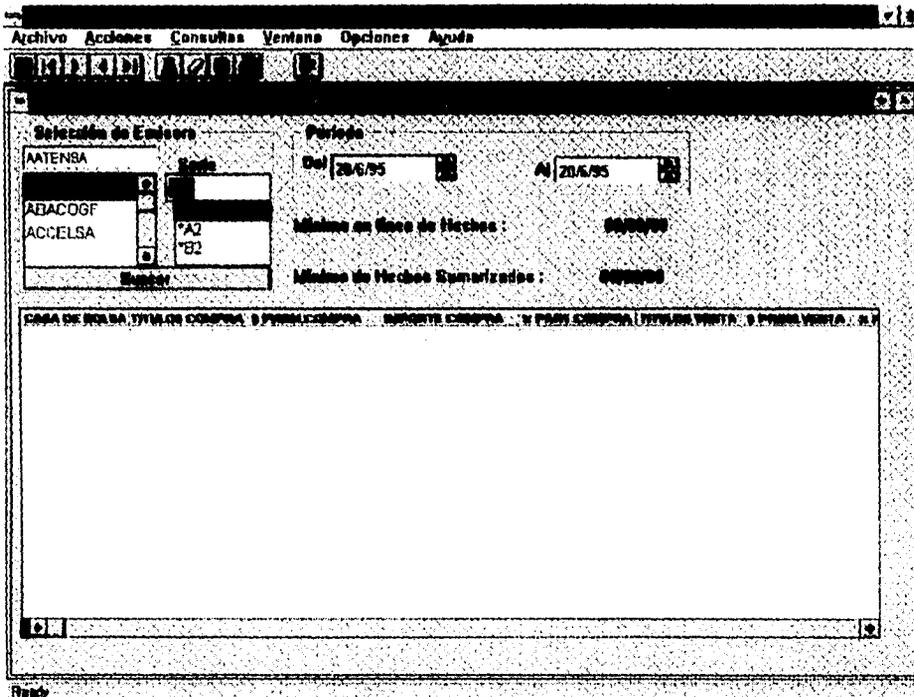
La información obtenida se encuentra ordenada por fecha y hora, calculándose los totales por fecha.

Tablas

Las tablas que se accesan para poder realizar esta consulta son :

- EMISION_RV
- INTERMEDIARIOS
- DETALLE_HECHOS_CAPITALES_RV

Ventana W_MOV_NETO_EMITORAS



Funcionamiento

Ventana que tiene como objetivo monitorear las operaciones que los intermediarios han realizado sobre una determinada Emisora-Serie en un periodo de tiempo determinado (por default se considera el día de hoy), con el fin de conocer a como han estado realizando las operaciones de compra y venta, así como poder determinar la posición en cuanto a títulos e importe que tienen sobre una determinada Emisora-Serie.

Esta consulta tiene asociada un **Procedimiento Almacenado** (Store Procedure) al momento de su ejecución, debido a que se realizan cálculos por cada intermediario.



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Casa de Bolsa
- Títulos que ha comprado
- Promedio de compra
- Importe comprado
- Porcentaje de participación de compra
- Títulos que se han vendido
- Promedio de venta
- Porcentaje de participación de venta
- Importe vendido
- Movimiento neto de títulos
- Movimiento neto de importe
- Porcentaje de participación en la serie
- Porcentaje de participación en capital

La información obtenida se encuentra ordenada por casa de bolsa

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- EMISION_RV
- INTERMEDIARIOS
- DETALLE_HECHOS_CAPITALES_RV
- SUMARIO_HECHOS_CAPITALES_RV_MEN



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Casa de Bolsa
- Títulos que ha comprado
- Promedio de compra
- Importe comprado
- Porcentaje de participación de compra
- Títulos que se han vendido
- Promedio de venta
- Porcentaje de participación de venta
- Importe vendido
- Movimiento neto de títulos
- Movimiento neto de importe
- Porcentaje de participación en la serie
- Porcentaje de participación en capital

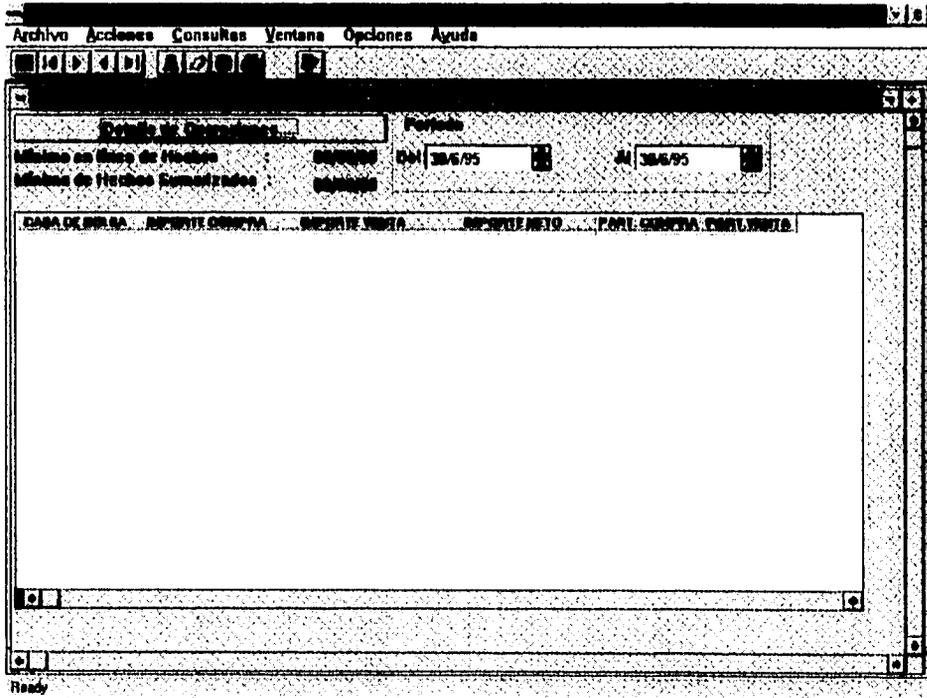
La información obtenida se encuentra ordenada por casa de bolsa

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- EMISION_RV
- INTERMEDIARIOS
- DETALLE_HECHOS_CAPITALES_RV
- SUMARIO_HECHOS_CAPITALES_RV_MEN

Ventana W_MOV_NETO_INTERMEDIARIO



W_MOV_NETO_INTERMEDIARIO

Funcionamiento

Ventana que muestra el resumen sobre las operaciones de los intermediarios autorizados para operar en el piso de remates, dentro de un periodo determinado (por default día de hoy). Una vez obtenido el resumen de cada intermediario, se puede seleccionar cualquiera de ellos, para conocer de manera detallada sus operaciones. Esta consulta al igual que la ventana anterior tiene asociada un Procedimiento Almacenado debido a que se realizan cálculos por obtener el resumen de lo operado por Casa de Bolsa.



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Casa de Bolsa
- Importe comprado
- Importe vendido
- Importe neto
- Porcentaje de participación comprado
- Porcentaje de participación vendido

La información obtenida se encuentra ordenada por Casa de Bolsa. Para conocer el detalle de una determinada Casa de Bolsa se requiere seleccionar una de ellas, abriéndose la ventana W_MOV_NETO_INTERMEDIARIO_D

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- EMISION_RV
- INTERMEDIARIOS
- DETALLE_HECHOS_CAPITALES_RV
- SUMARIO_HECHOS_CAPITALES_RV_MEN



Ventana

W_MOV_NETO_INTERMEDIARIO_D

EMPRESA	OTROS	TITULOS COMPRA	PRECIO COMPRA	IMPORTE COMPRA	TITULOS VENTA	PRECIO VENTA
				\$0.00		

W_MOV_NETO_INTERMEDIARIO_D

Funcionamiento

Ventana que tiene como objetivo conocer el detalle sobre lo operado de una casa de bolsa en particular dentro de un periodo determinado. Esta ventana es abierta desde la ventana W_MOV_NETO_INTERMEDIARIO_D al seleccionar una casa de bolsa y oprimir el botón de "Detalle".

Esta consulta al igual que la ventana anterior tiene asociada un Procedimiento almacenado debido a que se realizan cálculos por obtener el detalle de lo operado de una determinada Casa de Bolsa.

Información

La información que se obtiene de la base de datos , es desplegada en forma tabular y es la que a continuación se presenta:

- Emisora
- Serie
- Títulos que han comprado
- Promedio de compra
- Importe comprado
- Títulos que se han vendido
- Promedio de venta
- Porcentaje de participación de venta
- Importe vendido
- Movimiento neto de títulos
- Movimiento neto de importe

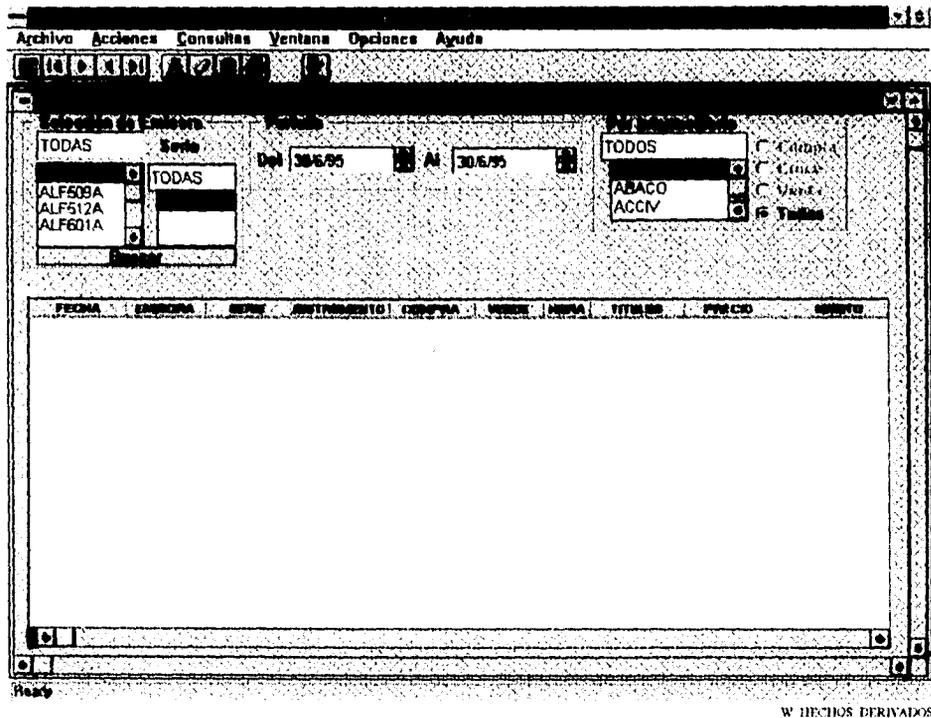
La información obtenida se encuentra ordenada por el movimiento neto de importe.

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- EMISION_RV
- INTERMEDIARIOS
- DETALLE_HECHOS_CAPITALES_RV
- SUMARIO_HECHOS_CAPITALES_RV_MEN

Ventana W_HECHOS_DERIVADOS



Funcionamiento

Ventana que es utilizada para consultar todos los hechos realizados sobre emisiones de productos derivados (emisiones de Renta Variable) en un periodo determinado (por default día de hoy), según ciertas condiciones que el usuario seleccione. Lo que se busca en esta ventana, es monitorear todos los hechos de las emisoras derivadas que se han realizado en el día o sobre alguna emisora en particular, o las operaciones que esta efectuando un determinado intermediario sobre alguna emisora, etc (La información obtenida dependerá del criterio de selección que se haya escogido).



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Fecha
- Emisora
- Serie
- Instrumento
- Comprador
- Vendedor
- Hora
- Títulos
- Precio
- Monto
- Tipo de Operación

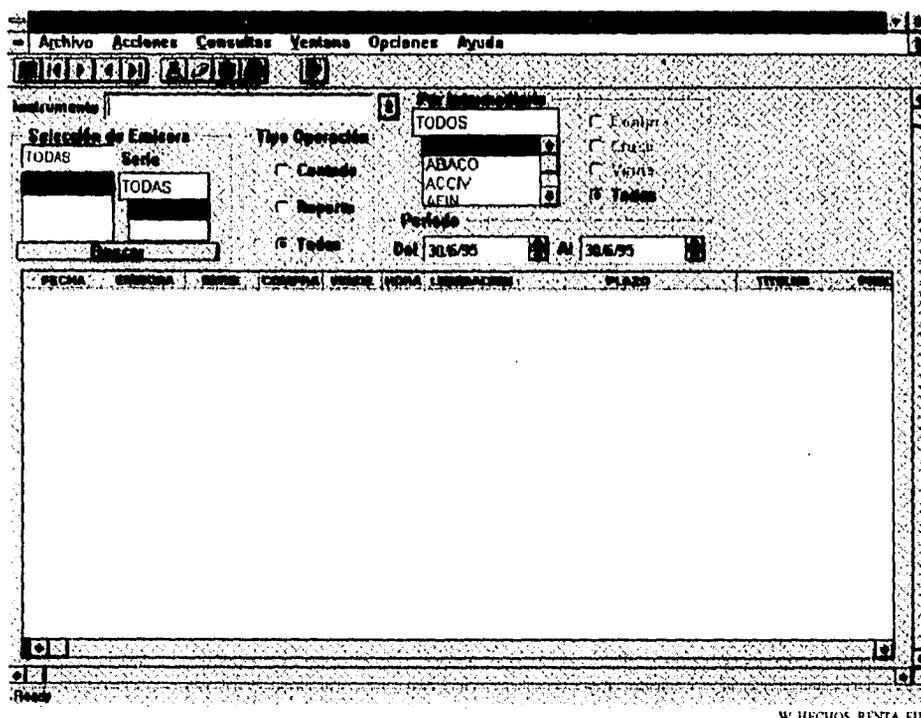
La información obtenida se encuentra ordenada por fecha, hora y se realizan totales por fecha.

Tablas

Las tablas que se accesan para poder realizar esta consulta son :

- EMISION_DERIVADOS
- INTERMEDIARIOS
- DETALLE_HECHOS_DERIVADOS

Ventana W_HECHOS_RENTA_FIJA



Funcionamiento

Ventana que es utilizada para consultar todos los hechos realizados sobre emisiones de Renta Fija en un periodo determinado (por default día de hoy), según ciertas condiciones que el usuario seleccione. A diferencia de las consultas de renta variable, la búsqueda de información es por instrumento y no por emisora, teniéndose que seleccionar primero un instrumento y apareciendo posteriormente las emisiones que manejan ese instrumento, seleccionando una o todas las emisiones. (La información obtenida dependerá del criterio de selección que se haya escogido).



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Fecha
- Emisora
- Serie
- Comprador
- Vendedor
- Hora
- Liquidación
- Plazo
- Títulos
- Precio
- Monto
- Tipo de Operación

La información obtenida se encuentra ordenada por fecha, hora y se realizan totales por fecha.

Tablas

Las tablas que se accesan para poder realizar esta consulta son :

- INSTRUMENTO
- EMISION_RF
- INTERMEDIARIOS
- DETALLE_HECHOS_RF

Ventana W_HECHOS_DINERO_INSTRUMENTO

The screenshot shows a software window titled "W_HECHOS_DINERO_INSTRUMENTO" with a menu bar (Archivo, Acciones, Consultas, Ventana, Opciones, Ayuda) and a toolbar. The main area contains a table of market data for CETES instruments. The table has columns for Instrumento, Tipo Operación, For Intermediario, and other details. The data rows show various transactions with values and dates.

Instrumento	Tipo Operación	For Intermediario	Valor	Fecha	Detalle
30/06/95 CETES	CETES	950824	MEXN	VALUE	13.20 MISMO DIA
30/06/95 CETES	CETES	950831	MEXN	VALUE	13.20 MISMO DIA
30/06/95 CETES	CETES	950907	MEXN	DE	12.26 MISMO DIA
30/06/95 CETES	CETES	950831	DE	MEXN	12.25 MISMO DIA
30/06/95 CETES	CETES	951221	INVER	BURSA	11.01 24 HORAS
30/06/95 CETES	CETES	950715	VALME	DE	10.48 MISMO DIA
Total de CETES		N.Hechos:		7	

Funcionamiento

Ventana que es utilizada para consultar todos los hechos realizados sobre instrumentos del mercado de dinero en un periodo determinado (por default día de hoy), según ciertas condiciones que el usuario seleccione. Al igual que en las consultas de Renta Fija, en las consultas sobre mercado de dinero, el instrumento es al que se le da seguimiento.

Lo que se busca en esta ventana, es monitorear todos los hechos de un instrumento del mercado de dinero que se han realizado en el día, o las operaciones que esta efectuando un determinado intermediario sobre algún instrumento (La información obtenida dependerá del criterio de selección que se haya escogido).



Información

La información que se obtiene de la base de datos, es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Fecha
- Instrumento
- Emisora
- Emisión
- Comprador
- Vendedor
- Hora
- Liquidación
- Plazo
- Banxico
- Tipo de Operación
- Títulos
- Precio
- Importe
- Tasa de Descuento
- Tasa de Rendimiento

La información obtenida se encuentra ordenada por fecha y hora, calculándose los totales por fecha.

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- INSTRUMENTO
- EMISION_DINERO
- INTERMEDIARIOS
- DETALLE_HECHOS_DINERO

Ventana W_HECHOS_DINERO_CLASIFICACION

Archivo Acciones Consultas Ventana Opciones Ayuda

Instrumentos: Gubernamentales Bancarios Privados

Liquidación: Contado Reporte Tesoro

Período: Del 30/6/95 Al 30/6/95

Por Intermediaria: TODOS ABACO ACCN AFIN Balcera

FECHA	INSTRUMENTO	ESPECIE	MONEDA	OPERACION	VALOR	MONEDA	LIQUIDACION
30/06/95	CETES	CETES	951214	INVER	PROBU	13.57	MISMO DIA
30/06/95	CETES	CETES	950024	MEXIN	VALUE	13.20	MISMO DIA
30/06/95	CETES	CETES	950031	MEXIN	VALUE	13.20	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960027	INVER	ACCN	12.32	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960002	INVER	ACCN	12.31	MISMO DIA
30/06/95	CETES	CETES	950907	MEXIN	IXE	12.26	MISMO DIA
30/06/95	CETES	CETES	950031	IXE	MEXIN	12.25	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960002	ACCN	ANAHU	12.10	MISMO DIA
30/06/95	AJUSTABONOS CINCO AÑOS	AJUBONO	960030	INVER	VECTO	12.06	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960002	ACCN	PROBU	11.54	MISMO DIA
30/06/95	AJUSTABONOS CINCO AÑOS	AJUBONO	960030	VECTO	INVER	11.53	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960425	CBMEX	IXE	11.47	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960028	CBMEX	IXE	11.46	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960425	CBMEX	MEXIN	11.22	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960425	IXE	MEXIN	11.20	MISMO DIA
30/06/95	BONDES DEL GOB. FEDERAL	BONDES	960028	IXE	MEXIN	11.20	MISMO DIA
30/06/95	AJUSTABONOS CINCO AÑOS	AJUBONO	970430	ACCN	CBI	11.11	MISMO DIA
30/06/95	AJUSTABONOS CINCO AÑOS	AJUBONO	970012	ACCN	CBI	11.09	MISMO DIA

Leer lista de Base de Datos

W_HECHOS_DINERO_CLASIFICACION

Funcionamiento

Ventana que es utilizada para consultar todos los hechos realizados sobre una clasificación de instrumentos del mercado de dinero (Gubernamentales, Bancarios, Privados) en un periodo determinado (por default día de hoy), según las condiciones que el usuario hay escogido

Lo que se busca en esta ventana, es monitorear todos los hechos de la clasificación seleccionada que se han realizado en el día, o las operaciones que esta efectuando un determinado intermediario sobre algún instrumento



Información

La información que se obtiene de la base de datos , es desplegada en forma tabular (por columnas), siendo ésta la siguiente:

- Fecha
- Instrumento
- Emisora
- Emisión
- Comprador
- Vendedor
- Hora
- Liquidación
- Plazo
- Banxico
- Tipo de Operación
- Titulos
- Precio
- Importe
- Tasa de Descuento
- Tasa de Rendimiento

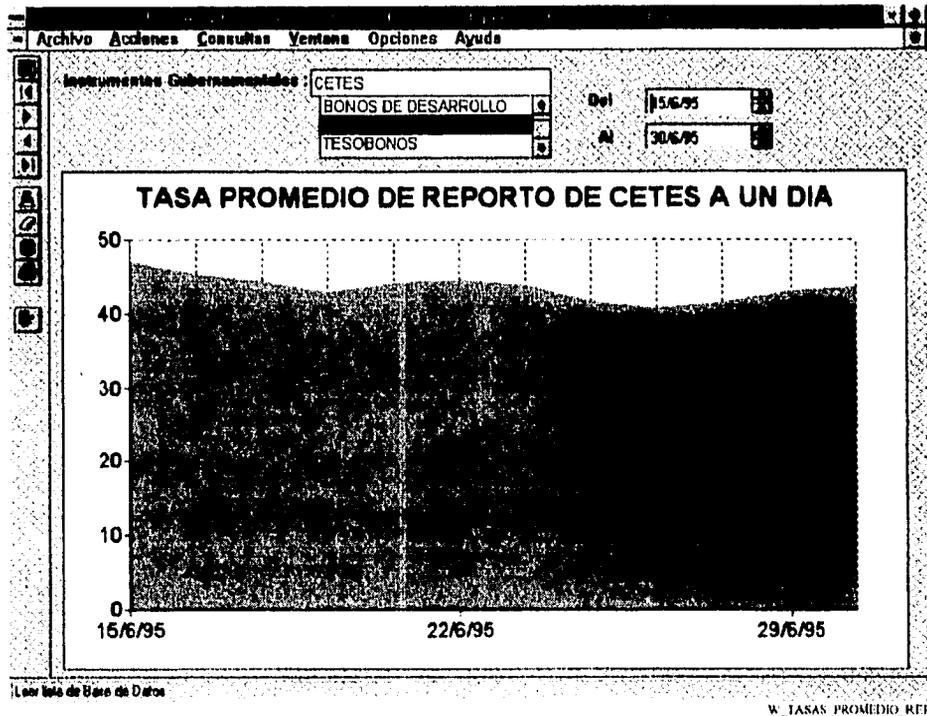
La información obtenida se encuentra ordenada por fecha y hora, calculándose los totales por fecha.

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- INSTRUMENTO
- EMISION_DINERO
- INTERMEDIARIOS
- DETALLE_HECHOS_DINERO

Ventana W_TASAS_PROMEDIO_REP



Funcionamiento

Ventana que tiene como objetivo conocer de una forma gráfica, el comportamiento que ha tenido la tasa promedio de los instrumentos que emite el Gobierno Federal (Cetes, Tesobonos, Ajustabonos, etc.) dentro de un periodo de tiempo determinado, utilizando para ello, la información de las operaciones (hechos) realizadas en el mercado secundario sobre los instrumentos del gobierno. Fué necesario realizar un **Stored Procedure** para el calculo de la tasa promedio



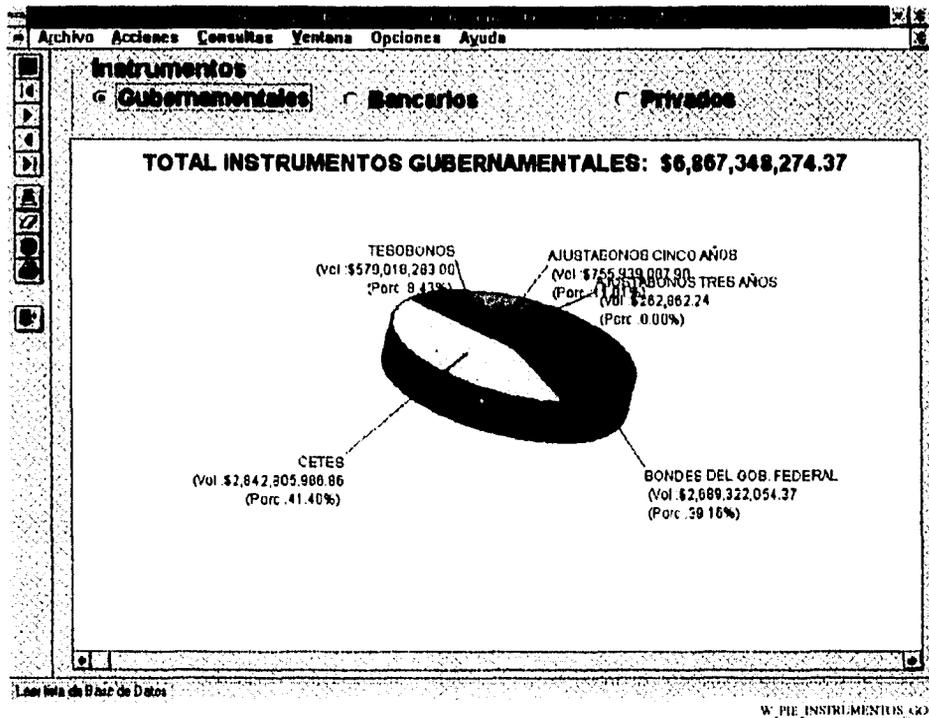
Información

La información que se obtiene de la base de datos , es desplegada en forma de gráfica siendo el primer punto en el eje de las x's la fecha "Del" y el último punto la fecha "Al" previamente seleccionadas. Para el eje de las y's se consideró la tasa promedio de operaciones de Cetes, siendo esta un dato calculado, ya que la Bolsa Mexicana de Valores no lo envía .

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- INSTRUMENTO
- DETALLE_HECHOS_DINERO
- HISTORICO_TASAS_MD



Funcionamiento

Ventana que tiene como objetivo conocer de una manera gráfica (pie), que porcentaje e importe de lo operado en el día le correspondió a cada instrumento dentro de la clasificación de instrumentos de mercado de dinero, calculándose el total de lo operado por clasificación (Gubernamentales, Bancarios y Privados), calculándose el total operado en el día.



Monitor de Mercados

Información

La información que se obtiene de la base de datos , es desplegada en forma de "PIE", siendo cada rebanada un instrumento según la clasificación seleccionada previamente.

Tablas

Las tablas que se accesan para poder realizar esta consulta son:

- INSTRUMENTO
- DETALLE_HECHOS_DINERO



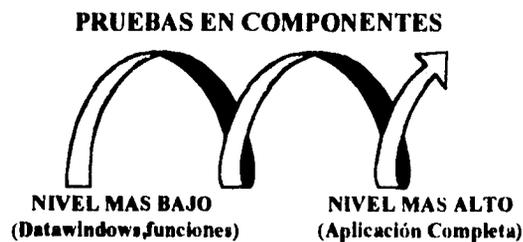
Pruebas

Conforme se va trabajando con herramientas de rápido desarrollo, es necesario probar y monitorear el comportamiento de determinado(s) componente(s) de la aplicación, esto con el fin de verificar, si el(los) componente(s) se comporta(n) de la manera esperada, antes de generar el código ejecutable, corrigiendo fallas de diseño y de construcción, para evitar de esta manera mala publicidad para la aplicación desarrollada.

PowerBuilder proporciona las herramientas necesarias para realizar pruebas y monitorear los diferentes componentes que formarán parte de una aplicación, contando con las siguientes facilidades:

- Permitir probar una aplicación, sin necesidad de generar un archivo ejecutable.
- Permite ver una ventana, datawindow, menú en el momento de su diseño (preview).
- Accesar información de la base de datos desde una datawindow.
- Cuenta con un rastreador de código (debugger).
- Compila código de un objeto a nivel evento del mismo.

Para esta aplicación, se aprovecharon estas facilidades realizándose las pruebas necesarias a nivel componente y corrigiéndose en ese momento las diferentes fallas detectadas.



Estas pruebas se realizaron, primero en los componentes de más bajo nivel (datawindow, funciones, menús, etc.) hasta probar el componente de más alto nivel (la aplicación completa), asegurándonos que bajo el ambiente de desarrollo la aplicación no tiene fallas.



Monitor de Mercados

Nota. - En la fase de Puesta en Marcha de la Aplicación, ya con el ejecutable creado y en un ambiente de producción, fue necesario realizar las pruebas necesarias, para verificar el acceso a la base de datos, la autenticidad de los datos obtenidos, despliegue de iconos y bitmaps, etc., verificando que todo funcione como se espera.



Puesta en marcha de la Aplicación

Creación del programa ejecutable

Crear un ejecutable de una aplicación es cuestión de empaquetar las diferentes piezas (objetos) que éste necesita de la manera más apropiada. Un ejecutable que se crea dentro de PowerBuilder puede estar compuesto por las siguientes piezas:

- Un archivo ejecutable
- Librerías dinámicas de PowerBuilder
- Recursos adicionales

Siempre se crea exactamente un ejecutable (archivo EXE) para cualquier aplicación de PowerBuilder que se quiera instalar., teniendo este archivo como mínimo, código que permite correr a la aplicación como una aplicación natural de la plataforma (Windows). Esto quiere decir, que para nuestro caso cuando los usuarios quieran arrancar la aplicación basta que den doble click sobre el archivo ejecutable desde el Program Manager o desde algun grupo especial que se haya creado donde se encuentren todas las aplicaciones desarrolladas.

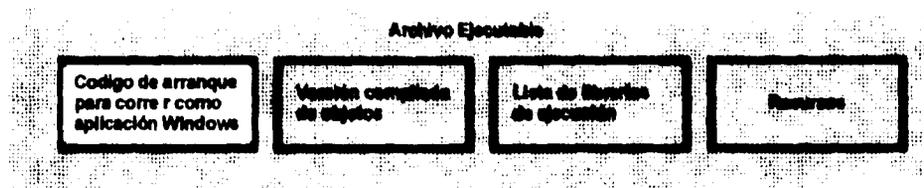
Dependiendo del modelo de empaquetamiento que se haya seleccionado, el archivo también contendrá uno o más de lo siguiente:

- Version Compilada de Objetos
Se puede escoger en poner todos los objetos dentro del archivo ejecutable (de tal forma que solo se tenga que distribuir un solo archivo) o en dividir la aplicación en un archivo ejecutable y en uno o más librerías dinámicas de PowerBuilder (archivos PBD)
- Lista de Librerías de Ejecución
Es una lista de librerías que el sistema de ejecución de PowerBuilder utiliza para encontrar objetos y recursos en cualquier librería dinámica empaquetada con anterioridad.
- Recursos
Se refiere a los diferentes recursos que la aplicación utiliza como bitmaps.



Monitor de Mercados

En esta figura se observan los diferentes bloques que puede tener un archivo ejecutable. Notese que el bloque que no está sombreado es el que siempre se tiene como mínimo y los demás bloques pueden estar o no dependiendo de la forma de empaquetamiento.



Para la aplicación desarrollada no se utilizó ni el segundo ni el último bloque de izquierda a derecha, utilizándose el manejo de librerías dinámicas. Las librerías dinámicas son parecidas a las DLL's, ya que se ligan a la aplicación a tiempo de ejecución.

Las librerías dinámicas pueden incluir recursos (bitmaps), cargándose estos en tiempo de ejecución. Los bloques que tienen las librerías dinámicas se muestran en la siguiente figura. El bloque que no se encuentra sombreado siempre existe.



Para la aplicación desarrollada se generaron dos librerías dinámicas (siab02.pbd y siab17.pbd), teniendo cada una de ellas el bloque de recursos.

Las razones por lo que se utilizaron librerías dinámicas en el desarrollo de esta aplicación a continuación se mencionan en la siguiente tabla.



Razón	Justificación
Modularidad	Disminuir el tamaño de la aplicación, teniendo archivos modulares más fáciles de manejar
Mantenible	Nos permite tener los componentes de la aplicación separados, esto con el fin de que si se quiere dar una nueva versión de la aplicación o existiera un error en la aplicación, no se tendría que redistribuir toda la aplicación, solo se actualizaría el archivo PBD particular.
Reusable	Para múltiples aplicaciones se pueden reusar componentes, esto es ya que los archivos PBD pueden ser compartidos entre aplicaciones.
Flexibilidad	Nos permite tener objetos dentro de la aplicación referenciados solamente dinámicamente en tiempo de ejecución
Eficiencia	Para una aplicación larga, como es este caso, se utiliza la memoria de manera más eficiente ya que: <ul style="list-style-type: none">◆ PowerBuilder no carga un todo el archivo PBD en memoria, solo carga objetos que va necesitando.◆ El archivo ejecutable se vuelve pequeño, haciendo su carga en memoria rápida.

Distribución de la aplicación a usuarios finales

La distribución de la versión ejecutable de nuestra aplicación a los usuarios finales, es básicamente cuestión de instalar todas las piezas adecuadas (archivo EXE, archivos PBD, archivos DLL, etc) en los lugares adecuados (en computados o en drives de la red). Esta instalación se dividió en dos partes:

- **Instalación de piezas de ambiente**
Se instalaron archivos DLL's (Dynamic Link Libraries), los cuales el usuario necesita para ejecutar cualquier aplicación de PowerBuilder. Estos archivos se instalaron en el directorio c:\siab\ de cada cliente (pc). La lista de las DLL's es:



Monitor de Mercados

PBGR040.DLL	PBDWE040.DLL	PBITXT40.DLL
PBCMP040.DLL	PBDWO040.DLL	PBLMI040.DLL
PBDBI040.DLL	PBECT040.DLL	PBOUI040.DLL
PBDBL040.DLL	PBIDBF40	PBPRT040.DLL
PBDEC040.DLL	PBVBX040.DLL	PBRTE040.DLL
PBTYP040.DLL	PBSHR040.DLL	PBRTF040.DLL

Se instaló también en el directorio c:\siab\ de cada cliente, el driver nativo de la base de datos de sybase 4.9.2, tendiendo el nombre de PBSYB040.DLL.

Se modificó el archivo WIN.INI de cada cliente, con el fin de que cualquier aplicación desarrollada al arrancar, encuentre las diferentes piezas de la aplicación (archivo ejecutable, archivos PBD, archivos DLL, etc) en una dirección dada, tecleandose las siguientes líneas de código:

```
[SIAB]
pathini=c:\siab\
```

Se verificó el acceso al servidor de base de datos, asegurandose de que cada cliente tuviera acceso al servidor.

- **Instalación de piezas de la aplicación**

Se instalaron en el drive c:\siab\, los siguientes archivos:

- SIAB02.EXE
- SIAB02.PBD
- SIAB17.PBD
- ARCHIVOS *.ICO, *.BMP
- SIAB02.INI

El archivo siab02.ini, es el archivo que contiene las variables de ambiente (base de datos a la se conectará la aplicación, dirección del archivo de ayuda, etc) que tendrá la aplicación.



Monitor de Mercados

Nota: Es importante considerar que paralelamente al desarrollo de la aplicación, todo un equipo de comunicaciones realizó la instalación de software (**NET-LIBRARY** y **OPEN CLIENT** para Sybase 4.9.2) para que cualquier cliente (pc) pudiera acceder la base de datos utilizando esta o cualquier otra aplicación.

Este software se instaló localmente en la dirección c:\sql de cada cliente , realizándose las pruebas necesarias para que existiera una conexión a la base de datos de Sybase 4.9.2 , para que de esta manera al llegar a esta fase del proyecto (Puesta en Marcha de la Aplicación), se realizará de una manera más dinámica y solamente se hicieran pruebas sobre la aplicación (que se abrieran las ventanas, que se dibujaran los iconos, que se prendieran o pagaran botones dependiendo del nivel de seguridad de cada usuario, que se accesará a la base de datos desde la aplicación, etc.), eliminándose considerablemente el tiempo de instalación en cada cliente.



Conclusiones

Es una realidad que el desarrollo de sistemas en un ambiente cliente/servidor es más complejo que en un ambiente tradicional, debido a que se requiere un amplio conocimiento en los diferentes componentes que abarca un esquema cliente/servidor (tales como desarrollo de aplicaciones utilizando herramientas front-end, manejo de la base de datos, manejo de los sistemas operativos tanto del cliente como del servidor, etc), mientras que el ambiente tradicional, es más sencillo, ya que todo está centralizado y es más homogéneo (un solo sistema operativo, las aplicaciones se desarrollan no utilizan ambiente gráfico, etc).

Es un hecho que la historia en cuanto a desarrollo de aplicaciones utilizando métodos y ambientes tradicionales no ha sido del todo buena y que como industria hemos estado muy lejos de las expectativas de nuestros clientes y que de seguir con ese tipo de resultados pronto quedaremos fuera del mercado. Para sobrevivir debemos:

- Reducir los tiempos de entrega de productos.
- Incrementar sustancialmente el porcentaje de sistemas utilizados al ser estos implementados y por lo tanto aumentar la calidad de nuestros sistemas.
- Instrumentar proyectos dentro de lo presupuestado y dentro del tiempo estimado.
- Reducir el porcentaje de recursos utilizados por proyectos.
- Reducir el porcentaje de recursos utilizados en mantenimiento.

Para lograr ello, necesitamos considerar una serie de acciones a seguir dentro del desarrollo de un sistema en ambiente cliente/servidor:

- Saber reclutar el personal adecuado para un proyecto.
Es importante contar con gente que sepa o que quiera aprender los diferentes componentes de un ambiente cliente/servidor (servidores de datos, herramientas front-end, sistemas operativos del cliente y del servidor, software de red, etc). De no contar con gente experta, considerar la opción de reclutar nueva gente, que transmite sus conocimientos a los demás miembros del equipo. Se tienen que definir claramente los diferentes roles de las personas involucradas en un proyecto.



Monitor de Mercados

- Utilizar metodologías incrementales (RAD)
El manejar este tipo de metodologías, permite la entrega de prototipos para que el usuario mida y entienda de una mejor manera el alcance de su sistema, disminuyendo con ello el tiempo en la entrega de productos a nuestros clientes.
- Manejar un modelo de datos
Utilizar una herramienta CASE para el modelaje de datos. Es importante considerar desde un principio tanto el modelo de los procesos que se llevaran a cabo en línea (OLTP) como el modelo para la toma de decisiones (DS).
- Manejar ambientes de desarrollo como ambientes de producción
El manejo de ambientes duplicados nos permite evitar la tentación de realizar cambios directamente en el ambiente de producción.
- Manejar ambientes de desarrollo como ambientes de producción
El manejo de ambientes duplicados nos permite evitar la tentación de realizar cambios directamente en el ambiente de producción.

Además de estos puntos mencionadas es importante considerar que como gente de sistemas es nuestra obligación tener un profundo conocimiento y experiencia en un determinado campo de acción como el sector financiero mexicano, manufactura, etc, ya que primero tenemos que entender el negocio de nuestros clientes para saber como aplicar cualquier tecnología para resolver los problemas de nuestros clientes.



Bibliografía

Libro/Revista	Autor
The Windows interface :An Application Design Guide	Microsfot Press
Object Oriented Technology : A manager Guide	David A. Taylor
Building Applications in PowerBuilder	Powersoft
DBMS Data Base Client Server Solutions , Client Server Deployment , "Do it right", Abril 1995	Robert C. Bolt
DBMS Data Base Client Server Solutions , C/S Developer, "4GL's: Productivity at What Cost ", Mayo 1994	David Lintchicum
DBMS Data Base Client Server Solutions , Enterprise Client Server , "Client/Server Culture Shock", Mayo 1995	Shaku Atrc
DBMS Data Base Client Server Solutions , "Following a Client/Server Database Methodology", Mayo 1995	David Shumberg
Las Nuevas Finanzas en México	Catherine Mansell Carstenes
Prontuario Bursátil y Financiero	Gonzalo Cortina Ortega



Glosario de Terminos Financieros

Acción

Titulo valor que representa una de las fracciones iguales en que se divide el capital social de una sociedad anónima. Sirve para acreditar y transmitir la calidad y los derechos de socio y su importe manifiesta el límite de la obligación que contrae el tenedor ante terceros y la empresa.

Agente de Bolsa (Intermediario)

Persona física autorizada por la Comisión Nacional de Valores y la Bolsa Mexicana de Valores para operar como intermediario en el mercado de valores. Es accionista de la Bolsa Mexicana de Valores y esta autorizado para efectuar la compra-venta de valores cotizados en Bolsa. Las casas de bolsa son agentes de Bolsa (Casa de Bolsa Bancomer, Accival, etc)

Banco de México (Banxico)

Banco Central y de emisión única. Organismo público descentralizado de la administración pública federal, con personalidad jurídica y patrimonios propios. Corresponde al banco de México desempeñar las siguientes funciones:

- Regular la emisión y circulación de la moneda y los cambios sobre el exterior, determinando el o los tipos de cambio a los que deba calcularse la equivalencia de la moneda nacional, para solventar obligaciones de pago en moneda extranjera, contraídas dentro o fuera de la República para ser cumplidas en ésta.
- Actuar como agente financiero del Gobierno Federal en las operaciones de crédito externo o interno y en la emisión y atención de empréstitos públicos, y encargarse del servicio de tesorería del propio Gobierno.
- Participar en representación del Gobierno, y con la garantía del mismo, en el Fondo Monetario Internacional.

Bolsa Mexicana de Valores

Estructura formal sobre la que descansan y dentro de la cual se desenvuelven las operaciones bursátiles en México.



Emisión

Acto de emitir valores de una empresa. Conjunto de valores que la emisora crea y pone en circulación (coloca). Una emisión puede ser pública si se "coloca" en el mercado bursátil, entre el gran público inversionista y a través de la intermediación bursátil (agente de valores), o privada si la colocación se hace entre los actuales accionistas o por la venta directa y privada (no entra el gran público inversionista)

Emisora

Empresa que pone sus valores a la venta a través de la Bolsa. Para recibir el calificativo de "emisora", se requiere que sea oferta abierta al gran público inversionista a través de la Bolsa.

Hecho

Sinónimo de transacción u operación bursátil. La palabra hecho se puede utilizar mediante expresiones como:

- ¿Cuál fue el último hecho de Telmex *L.?
- ¿Cuáles fueron los hechos de hoy?

Obligación

Título de crédito (tipo de instrumento) que representa la participación individual de sus tenedores en un crédito colectivo a cargo de una sociedad anónima. Por sus características existen diferentes tipos de obligaciones:

- Convertible
Obligación convertible en acción
- Indexada
Obligación que paga una tasa de interés indexada (dependiente de otro factor), como por ejemplo dos puntos arriba de la tasa primaria de Cetes a 28 días.
- Fiduciaria
Obligación que tiene como garantía un fideicomiso
- Hipotecaria
Esta garantizada con hipoteca sobre bienes propiedad de la sociedad emisora.



Monitor de Mercados

- **Prendaria**
Garantizada por diversos bienes
- **Quirografaria**
Garantizada unicamente por la firma de la emisora

Piso de Remates

Lugar específico en la Bolsa Mexicana de Valores, donde se lleva a cabo la realización de los remates de valores.

Precio

Valor monetario que se le da a un título en sus operaciones de compra - venta. El precio de "cierre", es el precio con el que cerro una emisión o instrumento en un día en que hubo transacciones con él. En caso en que no haya habido transacciones se tratara del último hecho.



Glosario de Terminos Técnicos

Application Painter

Utilería dentro de PowerBuilder que permite crear y definir los detalles (librerías, variables globales, icono, etc) que tendrá la aplicación.

Application Framework

Colección de clases (objetos ancestros) de donde se pueden heredar la mayoría de los objetos que se utilizarán para el desarrollo de una aplicación dentro de PowerBuilder.

CASE (Computer Aided Software Engineering)

Integración de disciplinas que utilizando un software, que soportado por una metodología, desarrollan y documentan sistemas.

DS (Decision Support) Model

Modelo de datos que se refiere a las tablas que se crean en un sistema para realizar consultas, agilizando el tiempo de respuesta de las mismas y quitando carga de trabajo al servidor.

DLL (Dynamic Link Library)

Librería que se crea utilizando el lenguaje de programación C y contienen funciones específicas, que son utilizadas en tiempo de ejecución.

Evento

Es cuando el usuario realiza una acción determinada sobre alguno de los objetos involucrados en la aplicación, como una ventana, menú, etc. Dependiendo de las características del objeto, este contará con eventos propios de él

Interfase de usuario (Graphical User Interface GUI's)

Es la comunicación visual que tiene un usuario final con algun sistema, siendo este último mucho más amigable para el usuario

Joint Application Development (JAD)

Técnica estructurada que su fuerza radica en formar grupos de trabajo para obtener, entender y validar requerimientos de clientes, pudiendo ser aplicada en diferentes puntos del proceso de desarrollo de un sistema (System Life Cycle).



Multiple Document Interface (MDI)

Es un estilo de ventana que se utiliza para abrir varias ventanas (hojas de trabajo) dentro de ella, teniendo el control la ventana MDI sobre las ventanas que se van abriendo. La mayoría de las aplicaciones de Windows son aplicaciones MDI (Word, Excel, Powerpoint).

Objeto

Conjunto de software empacado que maneja una colección tanto de datos relacionados (en forma de variables) como de procesos

OLTP (On Line Transaction Process) Model

Modelo de datos que se refiere a todas las tablas de tipo transaccional (altas, bajas y cambios) que contempla un sistema.

Procedimiento almacenado (Store Procedure)

Procedimiento que puede manejar una o varias instrucciones SQL con el fin de manipular información, se crean y se almacenan en la base de datos.

Rapid Application Development (RAD)

Metodología incremental que se utiliza para el desarrollo de aplicaciones dentro de la arquitectura cliente/servidor.

Script

Código procedural que se escribe utilizando el lenguaje de PowerBuilder (Powerscript Language) para expresar un procesamiento lógico a ejecutarse. Este código se escribe en algún evento de un determinado objeto.

Stored Procedure (Procedimiento Almacenado)

Procedimiento que se genera con estructuras de sql's y se guarda en la base de datos para ser utilizado posteriormente.

SQLCA (SQL Communication Area)

Objeto que crea PowerBuilder por default, el cual se utiliza para comunicarse con el servidor de base de datos, definiéndose en él diferentes atributos, como el nombre de la base de datos, el logid del usuario, el tipo de base de datos, etc



Monitor de Mercados

Toolbar

Botones asociados a una ventana de tal forma que el usuario en lugar de seleccionar una opción del menú simplemente da click sobre un botón determinado.