

92
2ej.



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

INFORMACION OPERATIVA EN LINEA

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N :
S A R A I L T I N O C O M U Ñ O Z
H E C T O R M O R E N O L I Z A R R A G A
J U A N M A R T I N C A S T R O R O J A S

DIRECTOR DE TESIS: ING. SALVADOR PEREZ VIRAMONTES



MEXICO, D. F.

1992

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está 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.

INDICE

	Pág.
PROLOGO	A
INTRODUCCION	1
CAPITULO I.	
CONCEPTOS FINANCIEROS.	
I.1 MERCADOS FINANCIEROS	5
I.1.1 INSTRUMENTOS BURSATILES A CORTO PLAZO	7
I.1.2 INSTRUMENTOS BURSATILES A LARGO PLAZO	10
I.1.3 INSTRUMENTO BANCARIOS.....	12
I.2 IMPORTANCIA DEL MANEJO DE LA POSICION, TASAS Y RESULTADOS	13
CAPITULO II.	
DISEÑO DE LA ESTRUCTURA DEL SISTEMA DE INFORMACION OPERATIVA EN LINEA.	
II.1 PLANEACION	16
II.1.1 DEFINICION DEL PROBLEMA	17
II.1.2. OBJETIVO	18
II.2 ANALISIS	18
II.2.1 ESPECIFICACIONES DE LOS DIAGRAMAS..	19
II.3 DISEÑO	24
CAPITULO III.	
COMUNICACIONES.	
III.1 CONCEPTOS DE COMUNICACIONES	26
III.1.1 DEFINICIONES	27
III.1.2 ASPECTOS GENERALES DE REDES LOCALES (LANS)	34
III.2 COMUNICACION LAN-MAINFRAME	36
III.3 CARACTERISTICAS DEL FLUJO DE INFORMACION.	
III.3.1 SERVERS PARA ACCESO A BASES DE DATOS	41
III.3.2 RECEPTOR DE DATOS	43
III.3.2.1 LIBRERIA ASYNC.TPU	45
III.4 REQUERIMIENTOS	48

**CAPITULO IV.
INTERFACE GRAFICA AL USUARIO.**

IV.1	CONCEPTOS GENERALES DE GRAFICACION	51
IV.2	DESARROLLO DE UNIDADES PARA GRAFICACION ..	54
IV.3	REQUERIMIENTOS	60

**CAPITULO V.
IMPLEMENTACION DEL SISTEMA DE INFORMACION
OPERATIVA EN LINEA.**

V.1	PROGRAMACION ORIENTADA A OBJETOS	62
V.2	ALGORITMOS DE EJECUCION	65
V.2.1	ALGORITMO DE POSICIONES	67
V.2.2	ALGORITMO DE TASAS	76
V.2.3	ALGORITMO DE RESULTADOS	83
V.3	PRESENTACION	93
V.3.1	DESARROLLO DEL MENU	93
V.3.2	ALGORITMO DEL MENU	96

CONCLUSIONES	99
--------------------	----

BIBLIOGRAFIA	102
--------------------	-----

PROLOGO

El motivo del presente trabajo es utilizar los conocimientos adquiridos durante el transcurso de la carrera de ingeniero en computación, para dar solución a problemas reales presentados en nuestra vida diaria.

Para la realización de dicho trabajo se investigaron las nuevas tecnologías y herramientas de programación para poder tener un sistema eficiente y con un tiempo de vida no muy corto.

Como el sistema tiene aplicación financiera, es necesario antes que nada conocer las bases del mercado financiero para poder realizar un sistema a la medida de las necesidades que el usuario desea. Es por ello que el primer capítulo de nuestra tesis la dedicamos a definir los conceptos necesarios para poder tener una visión clara del problema a resolver.

Una vez que conocemos como funciona el mercado financiero, el siguiente paso es analizar el problema, detectando los puntos críticos para poder definir una solución y elaborar el diseño de un sistema de cómputo que pueda satisfacer tales necesidades, el cual explicado en el segundo capítulo del presente trabajo.

Una vez definida la solución a través de un sistema automatizado, éste cuenta con una fase muy importante para su elaboración, la cual es la comunicación en tiempo real (en línea), y que sin ella el sistema no podría cumplir con su función. Tal fase es definida en el tercer capítulo.

El cuarto capítulo de la tesis es dedicado a definir los conceptos de graficación necesarios para posteriormente elaborar nuestras herramientas de graficación, las cuales no ayudarán a presentar la información de manera concisa y clara.

Finalmente en el último capítulo se define la técnica de programación para poder unir todos los elementos que conforman el sistema, teniendo así un sistema de cómputo que dará solución al problema planteado.

INTRODUCCION

Las satisfacción de las necesidades humanas se lleva a cabo cuando las personas están dispuestas a intercambiar dinero por un producto o servicio llamando a esto Mercado.

En ocasiones los mercados se conocen por el tipo de producto operado, en este caso hablaremos del Mercado Financiero que tiene como producto los Activos Financieros relacionando al conjunto de personas que están dispuestas a intercambiar sus recursos económicos o haberes con los que cuentan, ya que este intercambio satisface sus necesidades de dinero y capital.

Siempre hay personas que requieren de recursos para liquidar sus compromisos, proseguir con los programas que se han marcado, o bien establecer nuevas empresas, es decir requieren de dinero o capital; también existen personas que desean obtener un rendimiento por su dinero, para ello acuden a los mercados de dinero y capital.

Aquí viene la necesidad de un manejo de información confiable tanto para este tipo de personas como para los responsables de manejar la administración de su capital, en éste proceso se enfrenta diariamente una gran cantidad de información, y se requiere dedicar gran parte de tiempo para

su análisis evaluando la tendencia y conforme a ésta buscar soluciones a los posibles problemas, las cuales van desde las muy triviales hasta las altamente complejas, y que para resolverlas, lo más importante en estos casos es la percepción de información a tiempo, para que de esta manera se obtengan soluciones óptimas y oportunas.

El procedimiento para encontrar estas soluciones puede ser visto como un proceso de toma de decisiones, que es una parte esencial en las organizaciones modernas. Las decisiones financieras en este caso, se caracterizan por considerar principalmente tres factores:

- El dinero
- El tiempo
- El riesgo

Estas decisiones implican generalmente la aplicación de una considerable cantidad de información y criterios humanos basados en la información, antes de elegir alguna alternativa.

El problema consiste en percibir a tiempo los cambios, ya que estos demuestran la posible tendencia. El mejor inversionista del mundo será aquél con la mejor percepción de las situaciones y que, por supuesto, aplique su percepción en la toma de las decisiones correspondientes.

El tiempo es un elemento dentro del cual se presentan las alternativas y se desenvuelven las percepciones de los inversionistas. En el mundo de las inversiones, la vigilancia constante de las mismas es una cualidad muy valiosa. De esta manera es posible detectar oportunamente los cambios que se presentan y aprovecharlos al máximo.

En la actualidad el empleo de los avances tecnológicos y científicos en todas las áreas del conocimiento humano, han generado altos niveles de calidad en los servicios. En el ambiente financiero el nivel de calidad y disponibilidad en los servicios que ofrecen las compañías financieras, se han elevado enormemente, gracias a la aplicación de sistemas como éste, el que se mencionara más adelante y que es el objetivo de esta tesis.

Los promotores tienen la responsabilidad de colocar de manera oportuna y adecuada la posición de los diferentes instrumentos bancarios en el mercado financiero.

El monto de cada uno de los instrumentos de mercado tiene una gran importancia en su manejo durante el día, pues se realizan una serie de transacciones para la buena colocación de éstos en el mercado; y por ello es importante considerar todos los cambios que se llegan a registrar, en éste monto durante la captación.

Para poder observar y percibir la información de la mejor manera se han propuesto elementos visuales; los cuales pueden expresar de manera sencilla y clara la situación que se está investigando. Esto nos permite considerar todas las alternativas y cursos de acción posibles y así mismo sus tendencias, puesto que uno de los factores importantes que intervienen en la toma de decisiones es el tiempo.

Consideramos que todo el proceso se lleve a cabo totalmente en línea, para que la información que se perciba sea la más confiable y oportuna para una acertada decisión.

El sistema consiste principalmente en dos partes:

- 1.- La parte de comunicaciones, que se encarga de realizar el enlace y acceso a diferentes fuentes de información, en tiempo real.
- 2.- La interface gráfica con el usuario, que se encarga de presentar, mediante histogramas y/o tablas, la información obtenida de la comunicación.

CAPITULO I

CONCEPTOS FINANCIEROS

Capítulo I

CONCEPTOS FINANCIEROS

Para la comprensión de la funcionalidad del sistema de información operativa en línea es necesario introducirnos al mundo financiero, es por ello que como primer capítulo se exponen conceptos básicos, los cuales nos ayudarán a explicar mejor el sistema.

I.1.- Mercados Financieros

La economía nacional ha adquirido en los últimos años una configuración sustancialmente distinta a la que tuvo en las décadas precedentes, como consecuencia de la interacción de diversos factores tanto internos como externos que proporcionaron perspectivas de desarrollo nunca antes conocidas. El descubrimiento de nuestra riqueza energética ha transformado a México en una nación que requiere de instituciones e instrumentos idóneos para respaldar su dinámica de crecimiento.

Dentro de las alternativas existentes de financiamiento para las actividades productivas del país, el Mercado de Valores ha llegado a ocupar un lugar destacado; formando parte de un mercado más amplio, que es el constituido por el Sistema Financiero, al cual podemos definir como el conjunto orgánico de instituciones que generan, recogen, administran y orientan o dirigen, tanto el ahorro como la inversión, dentro de una unidad político-económica.

El Sistema Financiero Mexicano se constituye por estas partes:

EL MERCADO DE DINERO es la actividad crediticia que se realiza con operaciones concertadas a corto plazo. Las instituciones básicas que lo constituyen son los bancos de depósito y de ahorro. Los medios de pago de este mercado son los documentos comerciales a corto plazo, como por ejemplo los pagarés y las letras de cambio.

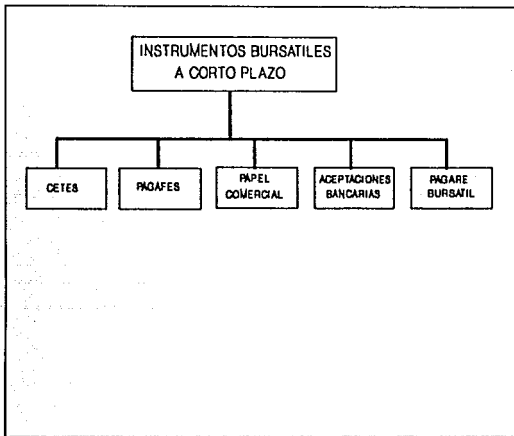
Por su parte, el MERCADO DE CAPITALES es el punto de concurrencia de fondos provenientes del ahorro de las personas, empresas y gobierno, con los demandantes de dichos fondos, que normalmente los solicitan para destinarlos a la formación de capital fijo.



Las instituciones básicas de este mercado son, por un lado, las sociedades financieras e hipotecarias, que conceden créditos a largo plazo; por otro lado, el Mercado de Valores al cual concurren directamente las empresas para colocar a través de la intermediaria financiera, sus acciones, obligaciones y otros instrumentos bursátiles, para ofrecerlos a los inversionistas.

Se puede definir al Mercado de Valores como el mecanismo que permite la emisión, colocación y distribución de los valores inscritos en el Registro Nacional de Valores e Intermediarios, a cargo de la Comisión Nacional de Valores y aprobados por la Bolsa Mexicana de Valores. La oferta en este Mercado esta formada por el conjunto de títulos emitidos por el sector empresarial tanto público como privado. La demanda esta constituida por los fondos disponibles para la inversión, tanto de personas físicas como morales.

I.1.1.-INSTRUMENTOS BURSATILES A CORTO PLAZO



CERTIFICADOS DE LA TESORERIA DE LA FEDERACION (CETES).

CETES son instrumentos emitidos por el Gobierno Federal, a plazos establecidos de 28,56,91 y 182 días, a través del Banco de México, mediante un sistema de subasta, a la cual concurre el intermediario bursátil en su conjunto (Bancos, casas de bolsa y aseguradoras). Estos instrumentos, además de ser vehículo de financiamiento público, cumplen con una importante función de regulación monetaria. El valor nominal de cada CETE es de 10,000 pesos. Se negocian a tasa de descuento, otorgando su tasa de rendimiento equivalente

En cuanto al tratamiento fiscal de estos instrumentos, las personas físicas están exentas del Impuesto sobre la Renta, mientras que para las personas morales los rendimientos son acumulables a sus ingresos. La tenencia de estos títulos es reservada a personas mexicanas.

PAGARES DE LA TESORERIA DE LA FEDERACION (PAGAFES).

Los PAGAFES cumplen funciones similares a los CETES, además de que fomentan el ahorro interno y brindan cobertura a sus tenedores ante contingencias cambiarias. Los PAGAFES tienen un valor nominal de 100 dólares americanos; sin embargo, la adquisición por parte de los inversionistas y la amortización por parte del Banco de México, se realiza en Moneda Nacional, considerando el tipo de cambio controlado para tal efecto. Los plazos de emisión son variables y similares a los que se presentan en CETES.

Por lo que toca al régimen fiscal, las normas en función son las mismas que rigen a los CETES.

PAPEL COMERCIAL.

El papel comercial es un instrumento de financiamiento de empresas, documentado mediante un pagaré con vencimiento fijo. Actualmente existen diferentes tipos de Papel Comercial, pero en esencia otorgan los mismos beneficios:

Papel Comercial Bursátil, Papel Comercial Quirográfico, Papel Comercial avalado por una Sociedad Nacional de Crédito, y Extrabursátil.

El Papel Comercial Bursátil es emitido exclusivamente por empresas que tienen cotizadas sus acciones en Bolsa.

El Papel Comercial Quirográfico es emitido por empresas inscritas en la sección de valores del Registro Nacional de Valores e Intermediarios de la Comisión Nacional de Valores.

El Papel Comercial avalado por una S.N.C. es, como su nombre lo indica el avalado por una de esas instituciones.

El Papel Comercial Extrabursátil es un pagaré emitido por una empresa que puede o no tener cotización en el mercado accionario de la Bolsa.

El Papel Comercial en cualquiera de sus modalidades constituye una fuente de recursos para empresas con escasez de efectivo que necesitan financiar su capital de trabajo y operaciones diarias. El plazo mínimo autorizado es de 15 días y el máximo es de 91 días. Las tasa de descuento y de rendimiento se determinan libremente, aunque son mayores que las tasas del CETE, al constituir el Papel Comercial un instrumento de mayor riesgo.

El régimen fiscal para personas físicas es la retención del 21% sobre los primeros 12 puntos, en tanto para las personas morales el rendimiento obtenido es acumularse a sus ingresos.

PAGARES DE LA TESORERIA DE LA FEDERACION (PAGAFES).

Los PAGAFES cumplen funciones similares a los CETES, además de que fomentan el ahorro interno y brindan cobertura a sus tenedores ante contingencias cambiarias. Los PAGAFES tienen un valor nominal de 100 dólares americanos; sin embargo, la adquisición por parte de los inversionistas y la amortización por parte del Banco de México, se realiza en Moneda Nacional, considerando el tipo de cambio controlado para tal efecto. Los plazos de emisión son variables y similares a los que se presentan en CETES.

Por lo que toca al régimen fiscal, las normas en función son las mismas que rigen a los CETES.

PAPEL COMERCIAL.

El papel comercial es un instrumento de financiamiento de empresas, documentado mediante un pagaré con vencimiento fijo. Actualmente existen diferentes tipos de Papel Comercial, pero en esencia otorgan los mismos beneficios:

Papel Comercial Bursátil, Papel Comercial Quirográfico, Papel Comercial avalado por una Sociedad Nacional de Crédito, y Extrabursátil.

El Papel Comercial Bursátil es emitido exclusivamente por empresas que tienen cotizadas sus acciones en Bolsa.

El Papel Comercial Quirográfico es emitido por empresas inscritas en la sección de valores del Registro Nacional de Valores e Intermediarios de la Comisión Nacional de Valores.

El Papel Comercial avalado por una S.N.C. es, como su nombre lo indica el avalado por una de esas instituciones.

El Papel Comercial Extrabursátil es un pagaré emitido por una empresa que puede o no tener cotización en el mercado accionario de la Bolsa.

El Papel Comercial en cualquiera de sus modalidades constituye una fuente de recursos para empresas con escasez de efectivo que necesitan financiar su capital de trabajo y operaciones diarias. El plazo mínimo autorizado es de 15 días y el máximo es de 91 días. Las tasa de descuento y, de rendimiento se determinan libremente, aunque son mayores que las tasas del CETE, al constituir el Papel Comercial un instrumento de mayor riesgo.

El régimen fiscal para personas físicas es la retención del 21% sobre los primeros 12 puntos, en tanto para las personas morales el rendimiento obtenido es acumularse a sus ingresos.

ACEPTACIONES BANCARIAS.

Las Aceptaciones Bancarias son letras de cambio emitidas por empresas con cargo a un banco que acepta la obligación de pagar.

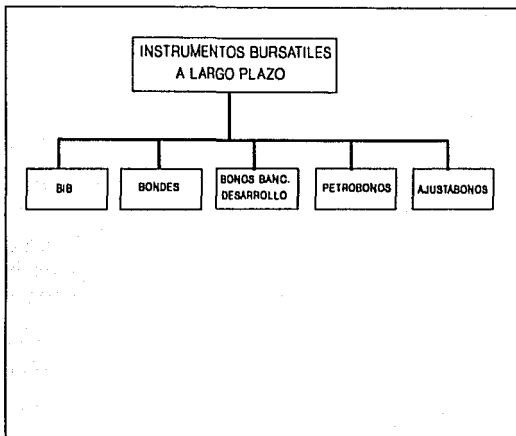
Este instrumento tiene por objetivo proporcionar financiamiento a empresas con necesidades de recurso a corto plazo, que por ser de menor calidad o por no estar registradas en bolsa no pueden emitir deuda directamente.

El rendimiento que ofrezcan las Aceptaciones Bancarias dependerá de la tasa de interés del mercado.

PAGARE BURSATIL.

Los Pagarés Bursátiles son títulos bancarios expedidos por instituciones a crédito a nombre del inversionista, quien desde el momento de su contratación, conoce la tasa de rendimiento y el importe total de los intereses que devengará su inversión y vencimiento.

I.1.2.-INSTRUMENTOS BURSATILES A LARGO PLAZO



BONOS DE INDEMNIZACION BANCARIA (BIB).

Los Bonos de Indemnización Bancaria son instrumentos que se emitieron para indemnizar a los accionistas de los bancos que se nacionalizados.

BONOS DE DESARROLLO (BONDES).

Los BONDES son instrumentos emitidos por el Gobierno Federal, como una estrategia que coadyuve a una planeación financiera de financiamiento a largo plazo. Estos instrumentos están diseñados para devengar intereses mensualmente con base en la tasa primaria de rendimiento de CETES a 28 días.

BONOS BANCARIOS DE DESARROLLO.

Son instrumentos de Renta Fija emitidos por las instituciones de banca de desarrollo autorizadas por las autoridades hacendarias. Los bonos tiene plazo mínimo de tres años, el cual tiene que comprender por los menos un año

de gracia, con amortizaciones mediante pagos semestrales, una vez cumplido el período de gracia.

PETROBONOS.

Los petrobonos son títulos emitidos por Nacional Financiera como sociedad fiduciaria, mediante el fideicomiso irrevocable constituido entre el Gobierno Federal de la Secretaría de Hacienda y Crédito Público y NAFINSA. Estos valores representan un derecho derivado de un contrato de compra venta de petróleo crudo con Petróleos Mexicanos.

BONOS AJUSTABLES DEL GOBIERNO FEDERAL (AJUSTABONOS).

Los AJUSTABONOS son títulos de crédito a largo plazo, denominados en moneda nacional, en los cuales se consigna la obligación directa e incondicional del Gobierno Federal de pagar una suma determinada de dinero.

Estos bonos proporcionan a los inversionistas cobertura contra el riesgo de erosión del valor real de sus ahorros, facilita a las aseguradoras del país la posibilidad de ofrecer seguros o tiempo, propicia el desarrollo de fondos de pensiones cuyo valor real no se deteriore con el transcurso del tiempo, propicia el desarrollo de fondos de pensiones y jubilaciones que conserven su valor real en favor de sus beneficiarios.

I.1.2.-INSTRUMENTOS BANCARIOS



DEPOSITOS RETIRABLES EN DIAS PREESTABLECIDOS.

Estos instrumentos presentan la conveniencia de ofrecer a sus clientes liquidez en sus ahorros.

PAGARES CON RENDIMIENTO LIQUIDABLE AL VENCIMIENTO.

Estos instrumentos, como su nombre lo indica, sólo pagan intereses al vencimiento. Su rendimiento se basa en la tasa de las inversiones a plazo fijo, reinvertida a la misma tasa. Estos instrumentos representan la única opción que existe actualmente en México de fijar una tasa de rendimiento a largo plazo.

INVERSIONES A PLAZO FIJO.

Las inversiones a plazo fijo (también llamadas CD's Bancarios) pagan intereses mensuales. Las tasas de interés que pagan (en forma mensual) son normalmente congruentes con las de los Pagares.

I.2.-IMPORTANCIA DE LAS POSICIONES, TASAS Y RESULTADOS.

El mundo financiero es muy interesante, es por ello que es necesario conocer las ventajas que ofrece éste, para obtener mejores resultados y ganancias.

El mercado financiero básicamente se divide en dos mercados :

Mercado de dinero.

Mercado de capitales.

En este nuestro caso, a nosotros nos interesa el mercado de capitales, ya que es el que maneja los instrumentos financieros.

La necesidad de tener liquidez del mercado, lleva a realizar ciertas operaciones dentro del mundo financiero.

Primero hay que obtener el capital suficiente para comenzar a trabajar, éste es captado a través de diversos centros financieros como lo son bancos y casas de bolsa.

Posteriormente hay que colocar éste dinero donde mejor convenga, para ello se tiene que hacer una estimación del número de clientes que se tendrán durante ese día, para ello se tienen clientes con contratos preestablecidos, así como clientes de otros bancos y casas de bolsa; también se debe hacer una estimación de la cantidad de dinero que se va a invertir en los diversos instrumentos financieros. Por que se tiene que tener cifras exactas, ya que si sobran instrumentos, éstos se vuelven pérdidas, o si sobra dinero, éste pierde valor al no invertirse para obtener ganancias; así que hay que tener el número de clientes con su respectivo capital que corresponda con el número de instrumentos disponibles en el mercado.

Todos los instrumentos financieros tienen un valor nominal dentro del mercado, el cual es fijado por las correspondientes autoridades, pero dependiendo de la demanda que se tenga de cada instrumento, éste va a cambiar su valor nominal, incrementándose o disminuyendo, dependiendo del caso, entonces su costo es llamado valor real.

Los instrumentos también tiene una tasa, la cual corresponde a las ganancias que se van a obtener por haber invertido en dicho instrumento. Esta tasa puede llegar hasta un máximo óptimo, el cual es el valor máximo que puede tener dicha tasa para que el instrumento tenga ganancias.

Las instituciones financieras con cobertura nacional, tienen el problema de colocar el dinero captado de la mejor manera posible para no tener pérdidas, no solamente en una entidad, si no en diversas entidades estatales, lo cual se convierte en un mercado complicado, ya que se debe balancear entre el dinero captado y el número de instrumentos disponibles a nivel nacional, en otras palabras, tener una buena posición por cada instrumento financiero.

Es por ello, que para estas instituciones es muy importante tener buenos resultados, esto es, tener buena liquidez, y para ello es necesario realizar buenas inversiones para tratar de aumentar las ganancias y disminuir las pérdidas.

Para obtener una buena posición nacional, es necesario conocer la posición de cada instrumento en cada Estado de la república en el momento adecuado, ya que si no se tiene esta información a tiempo se pueden perder ganancias.

Ya que puede suceder que, si en cierto lugar del país, se tiene dinero disponible para invertir, es necesario comprar instrumentos financieros, y no se tienen a la mano, es necesario recurrir a otras instituciones financieras para comprar papel y solucionar el problema, evitando que el dinero esté ocioso, pero teniendo que pagar más por el costo real de dicho papel. O por lo contrario, en otra parte del país se tiene papel de sobra, y para no quedarse con éste, se tiene que vender a bajo precio, disminuyendo su costo real.

En ambos casos se están teniendo pérdidas. Es por ello que el conocimiento de las posiciones de los instrumentos es de suma importancia para tener buenas inversiones, ya que se puede colocar el capital de los clientes en los lugares más adecuados e incrementar las ganancias.

Por ello surge la necesidad de tener información confiable, exacta y a tiempo de todas las entidades del país. Y es así como nace la idea de crear un sistema de cómputo que cubra éstas necesidades.

Tal sistema es motivo de esta tesis, y es llamado sistema de información operativa en línea. El cual se va a comunicar con los diferentes centros de procesamiento de datos del país y obtener la información requerida para la toma de decisiones.

El sistema de información operativa en línea cubre esta necesidad, ya que tiene una comunicación constante (en línea) con los centros de cómputo de todo el país, de esta forma se extrae el valor que tiene cada instrumento financiero en las diversas partes del país, posteriormente, esta información se concentra en una base de datos y se presentan en gráficas de histogramas, las cuales muestran el valor que va teniendo cada instrumento durante el día. Esto

es una gran ventaja, ya que éstas gráficas son más fáciles de interpretar que un listado con cifras numéricas que representen el valor de cada instrumento.

Además, con la ayuda de éstas gráficas se puede ver el comportamiento de los instrumentos y se pueden hacer estimaciones y pronósticos, lo cual ayuda mucho para la toma de decisiones en el mundo financiero.

El monitoreo de las posiciones, tasas y resultados es refrescado constantemente cada que el sistema detecta que hay cambios en las posiciones.

Otra parte del sistema, presenta los resultados obtenidos, esto es va mostrando la liquidez que se van obteniendo con las transacciones efectuadas, y apoya a los ejecutivos a decidir si las operaciones efectuadas son las correctas, o se tiene que cambiar de estrategia para mejorar los resultados.

CAPITULO II

DISEÑO DE LA ESTRUCTURA DEL SISTEMA INFORMACION OPERATIVA EN LINEA

Capítulo II

DISEÑO DE LA ESTRUCTURA DEL SISTEMA DE INFORMACION OPERATIVA EN LINEA

Este documento presenta las etapas que fueron aplicadas durante el desarrollo del sistema de información, que comprende varias fases : investigación, análisis, diseño, programación e implantación, todas estas fases nos condujeron a un incremento en la eficacia y eficiencia del diseño de este sistema.

II.1.-PLANEACION.

Para iniciar el desarrollo de este sistema nos fue necesario contar con un documento global que sirviera como fuente de consulta sobre todas y cada una de las acciones realizadas en el proyecto Informativo.

La detección de un problema y necesidad que requiere el usuario para la operación del sistema de información, dio como resultado que se realizara una solicitud para que esta a su vez originara una investigación que condujo a determinar la viabilidad de este proyecto.

La solicitud del usuario e investigación previa fue, pues, el punto de partida en este desarrollo que es la etapa en la que se estudia la factibilidad de solución o satisfacción del problema y necesidad.

II.1.1.-DEFINICION DEL PROBLEMA.

En el mundo financiero diariamente se maneja una gran cantidad de información que requiere de un análisis para tener una amplia percepción de la tendencia que tiene el mercado, ya que de esto depende el buen manejo del capital que se está invirtiendo, que corresponde al monto de los diferentes instrumentos, sus respectivas tasas, ganancias y pérdidas obtenidas durante el transcurso día.

Actualmente se tiene un sistema, el cual es alimentado por cada centro financiero constantemente mientras mantienen sus actividades. Tal información es utilizada por los operadores de la banca de inversión para poder colocar de manera adecuada el capital de los inversionistas de tal forma que se puedan obtener los mayores beneficios.

Dicho proceso se realiza a nivel nacional considerando 6 regiones :

Monterrey
Metropolitana (Ciudad de México)
Veracruz
Guadalajara
León
Hermosillo

En cada región, se desarrolla el mismo mecanismo de consultar al sistema para obtener información de la posición tasas y resultados de los Instrumentos bancarios para realizar una buena inversión.

Debido al constante dinamismo que se tiene en los centros financieros resulta tedioso estar consultando al sistema de información, ya que éste presenta sus resultados por medios de tablas con cifras numéricas, lo cual no es lo más óptimo para tomar decisiones, y sobretodo para observar la tendencia que llevan los instrumentos durante el transcurso del día es necesario tener varios listados con los resultados de las consultas del día, lo cual va complicando el proceso de la toma de decisiones.

Pero además de todo lo anterior, se requiere realizar inversiones en las diferentes regiones del país sin afectar negativamente al resto de las regiones asociadas. Lo cual lleva a realizar comparaciones con diferentes instrumentos, diferentes mercados y diferentes regiones del país, teniendo así una labor muy complicada para las personas que van a colocar el capital de inversión.

Todo lo anterior, nos lleva a la conclusión de crear un sistema que apoye a los operadores de los centros financieros a la toma de decisiones, para que su tarea no se vuelva complicada. Por lo que el sistema debe monitorear el comportamiento del mercado financiero de tal forma que cuando se requiera la información no lleve mucho tiempo el tomar la decisión, y lo más importante, que la información proporcionada sea confiable y esté a tiempo.

II.1.2.-OBJETIVO.

Este sistema tiene como objetivo agilizar el proceso de toma de decisiones de la banca de inversión, haciéndolo más funcional, minimizando el tiempo de ejecución, desarrollándolo en tiempo real (en línea), generando una representación histórica por medio del monitoreo y explotación de la información, que permita la comparación de las posiciones, tasas y resultados entre instrumentos bancarios y las diferentes regiones del país, teniendo así un campo de acción mucho más grande, realizando la mejor distribución de los instrumentos bancarios, ayudando en gran manera a la toma de decisiones óptimas en todas las regiones involucradas.

Para esto se requiere de una buena presentación gráfica como interface para el usuario, de manera que pueda ser mucho más claro su comportamiento.

II.2.-ANALISIS.

Una vez que se ha planeado y estudiado la factibilidad de solución, o satisfacción de este problema, es conveniente realizar un estudio a fondo del caso, para poder determinar todos los factores que el mismo involucra. A este estudio se le denomina *análisis*.

El *análisis* es la segunda fase o etapa en el proyecto, en la que teniendo la definición y objetivos del problema así como requerimientos se deben de determinar todos los flujos de información y procesos que interactúan o deben interactuar para la solución del problema o necesidad que se habló

anteriormente. El producto final de esta fase es un conjunto de Diagramas de Flujo de Datos acompañados de sus respectivos Diccionarios de Datos.

Con ello se pretende plasmar las relaciones existentes entre la información, sus orígenes, destinos y, los procesos que se le aplican, mediante la elaboración de gráficas, a efectos de contar con una herramienta objetiva para mostrar las interrelaciones de este sistema.

La documentación y representación gráfica del sistema en cuestión se presentará más adelante. Estos diagramas están compuestos de 4 clases de figuras o elementos típicos, ellos son:

Fuentes / Destinos
Flujos de Datos
Procesos
Almacenamiento de Datos.

II.2.1.-ESPECIFICACIONES DE LOS DIAGRAMAS.

FLUJO DE DATOS

En los diagramas mostrados indicamos el sentido en el cual fluyen los elementos de información (datos) mediante líneas dirigidas (flechas) y, además, indicamos cuales datos viajan a través de ellos.

El nombre propio es para identificar el flujo, y a su vez, al elemento de información que viaja a través de él.

FUENTES / DESTINOS

Representamos a un ente físico, generalmente EMISOR o RECEPTOR de elementos de información o datos, a través de figuras rectangulares con un nombre propio adentro.

PROCESOS

Para representar una o más actividades o funciones afines, que se ejercen sobre algún elemento de información. Estas figuras están identificadas con un nombre propio y una etiqueta dentro de ellas.

La etiqueta sirve como referencia para interrelacionar los distintos DFD's que compongan a un sistema.

ALMACENAMIENTO DE DATOS

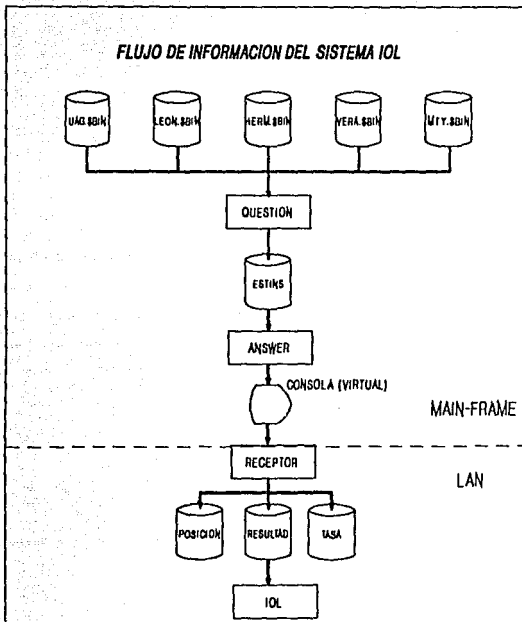
Sirven como depositarios, temporal o permanentemente, de elementos de información .

Son figuras rectangulares abiertas por el lado derecho.

DIAGRAMAS DE FLUJO DE DATOS

Existen dos diferentes tipos de diagramas de flujo, para ser explícitos, ellos son :

- Tipo cero o conceptuales (no tienen desglose).
- De detalle (estos pueden implicar uno o más niveles de desglose).



DICCIONARIO DE DATOS O DD (DATA DICTIONARY)

Para complementar la información usada en la representación gráfica del Sistema.

En el diccionario de datos se muestra la estructura de todos los elementos de cada archivo.

Además se define la información de cada uno de esos elementos.

Los elementos que emplearemos son las siguientes:

```
MaxInstrumentos = { Numero Máximo de instrumentos de
                   Mercado a manejar es igual a 15 };
MaxPlazos       = { Numero Máximo de plazos para las
                   diferentes Tasas };
Instrumento      = { Es el nombre del instrumento a
                   monitorear }
Costo           = { Es el costo de cada instrumento }
PosMax          = { Posición Máxima }
PosOpt          = { Posición Óptima }
CashMax         = { Tasa de Rendimiento Máxima de Cash }
CashOpt         = { Tasa de Rendimiento Óptima de Cash }
SpotMax         = { Tasa de Rendimiento Máxima de Spot }
SpotOpt         = { Tasa de Rendimiento Óptima de Spot }
Plazos          = { Numero de Plazos existentes }
Utilidad_Perdida = { Utilidad y Perdida }
```

DEFINICION DE LLAVES PRIMARIAS

```
Nodo : { Se refiere a la Región o zona nacional };
Hora : { Es el tiempo actual usado para el monitoreo
        de la información }
Tabla : { Definición de la tabla que contiene los
         instrumentos utilizados del Mercado Bancario }
}
```

ESTRUCTURA DE LA BASE DE DATOS

Registro de Posiciones:

Campo	Nombre	Tipo
1	Instrumento	String
2	Costo	Real
3	PosMax	Real
4	PosOpt	Real
5	CashMax	Real
6	CashOpt	Real

Registro de Tasas:

Campo	Nombre	Tipo
1	Instrumento	String
2	Plazos	Arreglo de registros

Registro de Plazos:

Campo	Nombre	Tipo
1	Plazo	Word
2	CashOpt	Real
3	CashMax	Real
4	SpotOpt	Real
5	SpotMax	Real

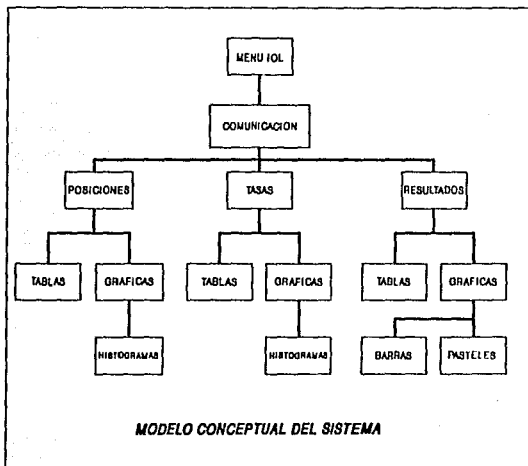
Registro de Resultados:

Campo	Nombre	Tipo
1	Instrumentos	String
2	Utilidad_Perdida	Real

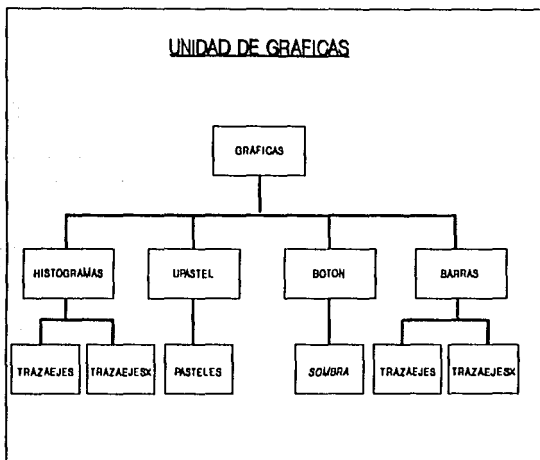
II.3.-DISEÑO.

Esta es la tercera etapa o fase del ciclo de vida de los sistemas de información. En ella se debe de construir el modelo que da solución o satisfacción al problema o necesidad, es decir, en el diseño es donde se transforma un Diagrama de Flujo de Datos en una Carta Estructurada o Diagrama Jerárquico, esta última es uno de los productos de la fase de diseño.

A continuación se muestra el diagrama conceptual del sistema, en el que se pueden observar los principales módulos que lo conforman, los cuales se detallan en capítulos posteriores.



Dentro del módulo de graficación se tienen varios subsistemas, los cuales se pueden visualizar en el siguiente diagrama :



CAPITULO III

COMUNICACIONES

Capítulo III.

COMUNICACIONES

III.1.-CONCEPTOS DE COMUNICACIONES.

El sistema de información operativa en línea está conformado por dos grandes módulos, de los cuales, el de comunicaciones es el motivo del presente capítulo. Dicho capítulo está dividido en cuatro partes :

En la primer parte se definen los términos y conceptos básicos necesarios para poder explicar el funcionamiento de los módulos de comunicación.

La segunda parte explica como es que se estableció la comunicación entre los sistemas de computadoras personales y el main frame, para poder acceder información de lugares remotos.

En la tercera parte se menciona el flujo de la información tanto en el computador central, como en la red local de microcomputadoras.

Finalmente en la última sección del capítulo se mencionan las herramientas necesarias para implementar las comunicaciones del sistema.

III.1.1.-DEFINICIONES.

RED DE COMPUTADORAS .- Es el conjunto de computadoras independientes enlazadas entre sí, con el fin de compartir recursos de cómputo.

CLASIFICACION DE REDES DE COMPUTADORAS POR SU EXTENSION GEOGRAFICA

Distancia entre Procesadores	Ubicada en	Nombre de la Red Extensión Geográfica
menos de 1 m.	Mismo Gabinete	Procesadores altamente Acoplados
1 m.	Mismo Gabinete	Sistema Multiprocesador
10 m.	Cuarto o sala	Local Area Network
100 m.	Edificio	Local Area Network
1000 m.	Campus	Local Area Network (LAN)
10 Km.	Ciudad	Wide Area Network (WAN)
100 Km.	País o Estado	Wide Area Network (WAN)
>1000 Km.	Continente	Interconexión de WANS

OBJETIVOS DE LAS REDES DE COMPUTADORAS

- Compartir Recursos de cómputo : procesadores, periféricos, programas, datos, etc.
- Agilizar la comunicación entre personas, programas, etc.
- Aumentar la capacidad de procesamiento distribuido.
- Proveer al usuario, dondequiera que se encuentre, con los recursos de cómputo e informáticos que requiera.
- Proveer compatibilidad entre hardware y software de diferentes fabricantes.

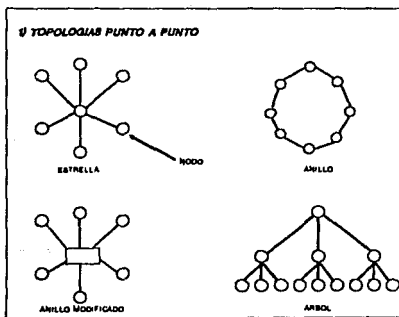
- Permitir, con relativa facilidad, una capacidad de crecimiento. (por ejemplo agregar CPUs).
- Permitir abaratar el costo de desarrollo de software, evitando sistemas de propósito general en una sola computadora.

TOPOLOGIAS DE REDES DE COMPUTADORAS

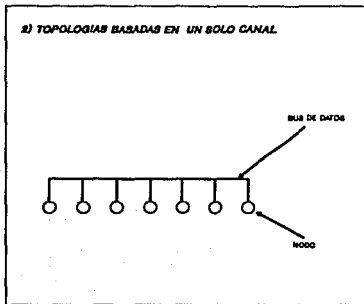
Topología en el campo de las redes de computadoras es la disposición física de los nodos o IMPs (Interface Message Processor) y sus enlaces.

Se distinguen dos tipos de topologías generales :

- 1) Topologías punto a punto.
- 2) Topologías basadas en un sólo canal (Broad Cast).



2) TOPOLOGIAS BASADAS EN UN SOLO CANAL



ARQUITECTURA DE LAS REDES

Arquitectura en términos de redes de computadoras es un conjunto de capas y protocolos que facilitan la implementación de las funciones en una red, así como la introducción de nuevas tecnologías a la red.

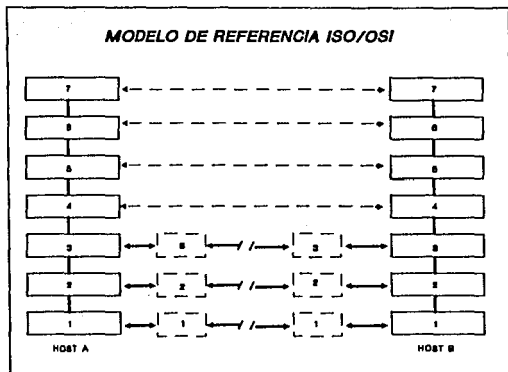
PROTOCOLO .- Conjunto de reglas de comunicación entre dos entidades.

CAPA .- Grupo de funciones perfectamente definidas que permite la adaptación de los datos para comunicación a través de un protocolo.

MODELO DE REFERENCIA ISO/OSI

ISO.- International Standares Organization. Organización dependiente de la ONU, la cual está encargada de establecer estándares de comunicación.

OSI.- Open System Interconection. Es el modelo general para arquitectura de redes de computadoras, el cual posee siete capas:



Capa 1. Física.

Define todos los equipos eléctricos, electrónicos, mecánicos y temporales que intervienen en la conexión de computadoras.
Se definen a nivel hardware.

Capa 2. Enlace de datos.

Su objetivo es la definición de los detalles de comunicación entre pares de

IMP --- IMP
IMP --- HOST

Mediante estructuras como frames de datos, para asegurar una transmisión confiable de frames de datos, se suele definir también el uso de los frames de reconocimiento (*acknowledge*).

Otra función de ésta capa es detectar y/o corregir errores de transmisión.

Los frames de reconocimiento sirven para regular productores rápidos de frames contra consumidores lentos.

Un frame está formado por la colección de bits que conforman datos y/o reconocimiento más los bits de corrección/detección de errores.

Regula el tipo de tráfico entre computadoras adyacentes, esto es determina si la comunicación será half o full duplex.

Se implementa en ROM.

Capa 3. Enrutamiento o de red.

Se encarga de llevar un paquete desde el HOST/IMP origen hasta el HOST/IMP destino.

Por tanto debe considerar algoritmos para resolver situaciones que :

- Determine la mejor ruta a seguir por un paquete.
- Evitar situaciones de congestión.
- Evitar/corregir deadlocks (atacamientos).

Suele decidir entre el sistema operativo y el driver (programa manejador de un dispositivo) del sistema operativo.

Capa 4. Transporte.

Establece los mecanismos para que dos computadoras HOST inicien, mantengan y terminen una comunicación.

Cada HOST posee por lo menos una estructura de datos conocida como

- Mailbox.
- Puerto.
- Socket.

La cual sirve para recibir/enviar mensajes de otras computadoras. También permite la multiplexación de varios canales de comunicación virtuales a través de uno o varios canales físicos.

Capa 5. Sesión.

Permite el acceso de usuarios desde HOST remotos.

Las sesiones remotas se establecen con :

- otras personas de la red.
- un proceso remoto.

La capa de sesión debe procurar mantener la transferencia de los servicios y fallas en la red.

Se implementa como servicios genéricos del sistema operativo.

Capa 6. Presentación.

El objetivo de esta capa es dar un grupo de funciones necesarias, más no indispensables, que permitan compartir los recursos de la red, tales como :

- archivos.
- terminales.
- etc.

Entre las funciones más comunes se tienen :

- Encriptamiento.
- Compresión de texto.
- Emuladores (protocolos de terminales virtuales)
- Protocolo de transferencia de archivos.

Capa 7. Aplicación.

Se deja abierta para que el usuario ejecute diversos programas, ya sean desarrollados por el mismo o por otra compañía.

Productos típicos de esta capa son :

- + Bases de Datos.**
- + Sistemas Operativos.**

III.1.2.-ASPECTOS GENERALES DE REDES LOCALES (LOCAL AREA NETWORKS LANs)

La necesidad de disponer de información, no solo exacta, sino además rápida en las empresas, en unión de la aparición de las computadoras personales, llevó al nacimiento de las redes de comunicación de datos en área geográfica reducida.

Una red de área local se ha convertido en una de las herramientas más importantes, y con una proyección de futuro aún mayor, para el intercambio de información, la realización de tareas de forma distribuida en los diferentes ámbitos de trabajo y la planificación de tareas.

Las Redes Locales de Microcomputadoras (LANs) se están extendiendo rápidamente, debido a que ofrecen la misma potencialidad que las Minicomputadoras y Mainframes pero a una fracción del costo de éstas.

Una red de área local es un conjunto de elementos físicos, que en coordinación, permiten compartir recursos hardware de alto precio, y prestaciones optimizando las inversiones de una empresa. Estos elementos deben estar interconectados y para realizar las tareas que se les encomiendan deben conversar entre sí. Los elementos lógicos que se integran en la red, son los que permitirán el intercambio de datos y por lo tanto la realización de las tareas de la red.

Las LANs se distinguen por :

- El área ocupada por la red es típicamente no mayor de un par de kilómetros.
- Tasa de transmisión mayor a 1 MB/s.
- Redes de Computadoras privadas.

Características secundarias :

- Topologías Regulares.
- Los medios típicos por los cuales se transmite son :
 - Cable trenzado.
 - Cable coaxial.
 - Fibra óptica.

Consideraciones para LANs :

- Ancho de banda, su aprovechamiento no es tan crítico como cuando existen enlaces compartidos (e.g. cable telefónico).
- Generalmente, el medio está casi libre de errores .
- Los nodos tienen gran facilidad para censar el medio (canal).

Actualmente se pueden escoger entre 20 Sistemas Operativos para Redes Locales, los cuales trabajan con herramientas de más de 60 distribuidores de ARCNET, 50 distribuidores de ETHERNET y 20 distribuidores de TOKEN-RING.

III.2.-COMUNICACION LAN-MAINFRAME.

Una red local es un sistema de comunicación y transmisión de datos que permite a un número de dispositivos físicos independientes intercambiar información con una probabilidad de error pequeña. Las características que diferencian una LAN (Local Area Network) de una red de área extendida WAN (Wide Area Network) son :

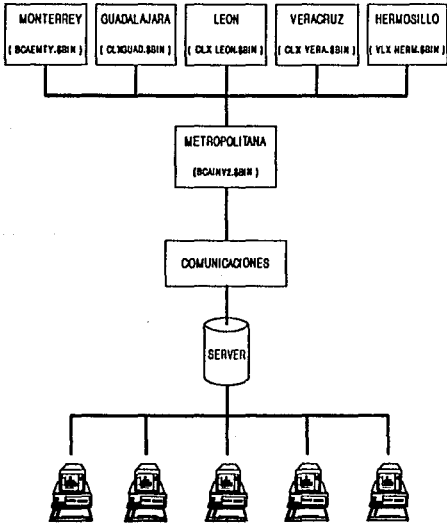
- El área geográfica en la que se extiende la red.
- La velocidad media de transmisión.

Esta última característica, cada vez menos significativa, pues los medios de transmisión son más eficientes y su costo disminuye. Existen en la actualidad, redes diseñadas a partir de las características de alta velocidad y baja pérdidas inherentes a la fibra óptica, que alcanzan velocidades de transmisión de hasta 100 Megabits por segundo.

Ya en la década de los 90 el desarrollo de redes de comunicación de datos parece abocada a un rápido crecimiento. Por una parte el software de las redes es más eficiente gracias a las mejoras en la seguridad de la red, los diseños tolerantes a fallas y al perfeccionamiento en las herramientas de gestión de la red.

La estandarización de las tecnologías utilizadas permiten la Interconexión de productos de fabricantes diversos, liberando a los usuarios de la dependencia del fabricante y la compatibilidad entre diversos sistemas.

FLUJO DE INFORMACION LAN-MAINFRAME



Considerando el modelo de tiempo compartido de computación utilizado por sistemas mainframes y minicomputadoras. Aplicaciones y datos son almacenados en una máquina central llamada host. El host provee la inteligencia para correr aplicaciones y procesar datos. Terminales tontas son usadas para acceder al host, su poder de procesamiento e información almacenada. Este modelo tiene definitivamente ventajas :

- Alto poder de procesamiento.
- Los datos son almacenados y controlados por una entidad central.
- Integridad altamente protegida.
- Sistemas operativos sofisticados para garantizar su confiabilidad.

Para implementar el SISTEMA DE INFORMACION OPERATIVA EN LINEA es necesario tener un sistema de cómputo capaz, no solo de soportar la comunicación con los centros de información en distintas regiones del país, sino además también mantener dicha comunicación en línea, soportar la transmisión de un gran flujo de datos y capacidad para procesarlos, y al mismo tiempo que la interface gráfica con el usuario sea de buena calidad.

Tomando en cuenta lo anterior, para realizar las comunicaciones con todo el país, almacenar y procesar los datos; se utilizará una computadora TANDEM VLX, la cual tiene una infraestructura que permite dialogar con otras computadoras TANDEM ubicadas en diferentes localidades, de una manera muy óptima y segura, ya que trabaja con la filosofía NON-STOP, para que la comunicación nunca sea interrumpida.

Con lo cual se podrán tener los beneficios de una WAN, ya que compartiremos recursos de cómputo de diversos puntos de la República, en cualquier momento.

Por otro lado para realizar la interface gráfica es necesario utilizar computadoras personales, ya que su tecnología permite implementar pantallas con excelente presentación, utilizando gráficas en tercera dimensión a colores, etc. Pero además de la interface gráfica, es necesario que varios usuarios accedan datos al mismo tiempo, es por ello que el SISTEMA DE INFORMACION OPERATIVA EN LINEA será implementado y ejecutado en Redes Locales de Microcomputadoras, ya que se aprovechan todas las características de una PC y al mismo tiempo uno a varios usuarios pueden acceder la misma información concentrada en el servidor de archivos que controla a dicha red.

Ahora bien, por un lado tenemos la comunicación y obtención de los datos de diferentes centros de información a través de computadoras TANDEM, y por otro lado se tiene la concentración de datos y la interface gráfica en una LAN; por lo tanto, es necesario establecer una comunicación entre la LAN y el MAINFRAME, para sumar las ventajas de estos dos sistemas de cómputo, y obtener así un tercer sistema con las características necesarias para implementar el SISTEMA DE INFORMACION OPERATIVA EN LINEA.

Por lo tanto tendremos una arquitectura cliente-servidor, la cual combina las características de los dos modelos mencionados anteriormente : el modelo de tiempo compartido de mainframes y minicomputadoras, y el modelo de recursos compartidos de redes locales.

Para establecer la comunicación entre la LAN y el MAINFRAME, fue necesario crear un programa para las microcomputadoras, el cual contiene una unidad de funciones que utilizan el puerto serial como medio de comunicación; y un programa receptor que utilice éstas funciones para acceder a la base de datos del MAINFRAME, extraiga la información deseada y la concentre en el área de almacenamiento del *file server*, que controla a la Red Local.

Por el otro lado, en la computadora TANDEM, es necesario crear una serie de programas llamados "*SERVEF*", los cuales tendrán acceso a las bases de datos de los distintos centros de información localizados en el interior de la República. Se concentrará toda la información deseada en un archivo, el cual esta disponible para ser enviado cuando se establezca la comunicación LAN-MAINFRAME.

El programa receptor de la microcomputadora dialogará con un programa *server* del mainframe, logrando la comunicación LAN-MAINFRAME. El primero le proporcionará las instrucciones necesarias al *server* para que éste último accese a la base de datos y extraiga de TANDEM la información deseada.

Finalmente el receptor leerá la información que le proporciona el *server* y la guardará en el disco del servidor de archivos de la red local, para que posteriormente pueda ser manipulada por cualquier estación de trabajo de la red local a través de nuestro sistema. De ésta forma, cualquier computadora de la red local puede obtener datos de cualquier centro de información del país al momento y con una interface amigable.

III.3.-CARACTERISTICAS DEL FLUJO DE INFORMACION.

En esta sección del capítulo de comunicaciones se explica donde se encuentran las bases de datos, como es que se accesan éstas y como es que la información llega hasta las computadoras personales para su presentación final.

III.3.1.-SERVERS PARA ACCESO A LA BASE DE DATOS.

La forma en que trabajan los programas en la computadora TANDEM, es dividiendo éstos en "*servers*" y "*requesters*".

Los programas server, son los programas que realizan funciones de lectura y escritura sobre los archivos de datos.

Los programas requester, son los programas que realizan una interface con los usuarios, validan datos de entrada y realizan una petición de lectura o escritura sobre los archivos de datos almacenados en computadoras TANDEM.

Los programas requester, realizan pantallas de presentación para que los usuarios puedan consultar, borrar, agragar o modificar datos almacenados en archivos. Los programas requester se comunican con los programas server para pasarle los requerimientos del usuario. Estos últimos atienden dichas peticiones, realizan los accesos a disco y devuelven al requester los resultados de su petición.

Con esto, en las computadoras personales, podemos hacer validaciones y pantallas de presentación al usuario. Con estos programas el usuario puede pedir las consultas que el desee en forma fácil y agradable para realizar sus decisiones. Estos programas se comunican con los servers de la computadora TANDEM, para que extraigan la información necesaria, pasarla al programa receptor de la PC, y finalmente sean utilizados para presentar gráficas que muestren el comportamiento de los instrumentos bancarios.

A continuación se describirán los programas "Server" de la computadora TANDEM, utilizados para tener acceso a las bases de datos distribuidas.

SERVER QUESTION.

Este programa es utilizado para leer la datos generados en diferentes regiones del país y que son almacenados en diferentes archivos, resume la información y la almacena en un archivo. Posteriormente éste archivo va a ser leído por otro server a petición de un programa receptor, el cual corre en una computadora personal.

El archivo donde se va a almacenar el resumen del comportamiento de los instrumentos bancarios de las diferentes regiones del país es : ESTINST (ESTado de la posición de los INSTRumentos).

Los archivos que van a ser leídos y que tienen la información del estado actual de los instrumentos bancarios por región son :

MEX-ESTINST : almacena el comportamiento de los instrumentos bancarios en el área metropolitana de la ciudad de México.

MTY-ESTINST : almacena el comportamiento de los instrumentos bancarios en Monterrey.

HER-ESTINST : almacena el comportamiento de los instrumentos bancarios en Hermosillo.

GUA-ESTINST : almacena el comportamiento de los instrumentos bancarios en Guadalajara.

LEO-ESTINST : almacena el comportamiento de los instrumentos bancarios en León.

VER-ESTINST : almacena el comportamiento de los instrumentos bancarios en Veracruz.

Estos archivos están indexados para poder accederlos dinámicamente.

SERVER ANSWER.

Es utilizado para leer el archivo ESTINS, que esta almacenado en la TANDEM.

Este programa es llamado para ser ejecutado a través de un programa llamado receptor, el cual es ejecutado desde una computadora personal que realiza la función de un gateway.

El programa receptor le proporciona las instrucciones al server answer, para que éste último haga las correspondientes lecturas al archivo ESTINS, y mande el resultado de la lectura a una consola, para que posteriormente el receptor lea dicha consola y guarde la información leída en una base de datos almacenada en el disco duro del servidor de la red local de microcomputadoras.

III.3.2.-RECEPTOR DE DATOS.

Una parte muy importante en la comunicación del sistema de información operativa en línea, es la parte en la cual el main-frame se comunica con los sistemas personales.

La parte que se encarga de realizar dicha comunicación del lado de las computadoras personales, es el programa *receptor*, el cual va a dialogar con el programa *answer* del sistema main-frame y extraer la información necesario para que el sistema de información operativa en línea pueda ser alimentado y cumpla con su función definida anteriormente.

El programa receptor realiza las capas 4 (transporte) y 5 (sesión) del modelo de referencia ISO-OSI para comunicación entre computadoras.

Para poder realizar tales funciones, el receptor se apoya en una librería de funciones comunicación construída anteriormente, la cual es llamada **ASYNC**. Tal librería cuenta con funciones para inicializar y programar el puerto serial de las computadoras personales, para utilizarlo como medio de comunicación; así como otras utilizadas para enviar y recibir mensajes por dicho puerto.

Estas funciones ayudan al receptor a establecer la capa de transporte del modelo de comunicación ya mencionado, ya que a través de ellas se mandan las instrucciones necesarias para rutear al puerto serial a una interface RS-232 y poder realizar la comunicación a la computadora TANDEM.

Una vez establecida la comunicación, el receptor envía los comandos necesarios para iniciar la sesión en la computadora TANDEM.

Una vez que la TANDEM, establece la sesión, el receptor esta listo para entablar el dialogo con el *server answer* y poder extraer la información deseada.

Ahora es cuando el receptor a través del puerto serie envía las instrucciones necesarias para que la computadora TANDEM ejecute el *server answer*, el cual extrae la información del archivo ESTINS y la deja en la consola lista para que el receptor la pueda leer.

A continuación el receptor comienza leer los datos que se encuentran en la consola y los envía a la base de datos de la red local, para que posteriormente el sistema de información operativa en línea pueda utilizarla para mostrar su comportamiento.

El programa receptor deja pasar un tiempo, el cual es variable y puede ser modificado dependiendo de las necesidades del usuario, antes de que vuelva a comunicarse con el *server answer* y extraer nuevos datos.

Una vez que paso dicho tiempo, el receptor lee la nueva información del archivo de la TANDEM, la cual fue previamente comparada por el *server answer* para evitar traer información redundante, agilizando el transporte de información.

Este último proceso es realizado constantemente mientras que hay actividades en el mercado financiero. Y cuando finalizan dichas actividades, se le indica al receptor, y este manda los correspondientes comandos a la computadora TANDEM para que detenga el *server answer*, cierre el archivo ESTINS y finalice la sesión de trabajo.

Además de la rutina mencionada anteriormente, el programa receptor cuenta con rutinas de contingencia para los casos de interrupción de la sesión trabajo, teniendo así información consistente y evitando pérdida o deformación de ésta.

I.I.I.3.2.1.-LIBRERIA ASYNC.TPU

Esta unidad es utilizada para inicializar, establecer y mantener comunicación a través del puerto serie de una microcomputadora. Es muy importante, ya que a través de los procedimientos y funciones aquí definidos, y con la ayuda del software de PCT (emulador de terminal TANDEM en PC), es como se realiza la comunicación entre una microcomputadora y el mainframe.

Entre los procedimientos más importantes contenidos en esta unidad se encuentran los siguientes :

INITPORT. Para inicializar todas las variables que se van a ocuparse durante la comunicación a través del puerto serie.

OPENPORT. Para preparar el puerto de serie por el cual se va a establecer la comunicación. Para ello deban pasarle ciertos parámetros como :

COMPORT : Define el puerto asíncrono que va a ser utilizado, el valor default que es utilizado es el 1, el cual corresponde al COM1

BAUDRATE : Define la velocidad de transmisión de datos con la cual se va a trabajar, en este caso en especial trabajamos a 9600 Bauds.

PARITY : Define el tipo de paridad que va a ser utilizada para confirmar que los datos fueron bien recibidos. En nuestro caso utilizamos paridad par.

WORDSIZE : Se le indica al puerto el tamaño que va a tener cada palabra a transmitirse, en este caso el tamaño de la palabra es de 8 bits.

STOPBITS : Indica el número de bits de paro necesarios para la transmisión de datos.

Cuando se enciende la computadora el BIOS inicializa la tabla `Async_BIOS_Port_Table`, almacenando en ella la dirección base de los adaptadores asíncronos instalados. Si se guarda un valor 0, es que no se ha inicializado ningún adaptador asíncrono.

Para comenzar la función, primero se pregunta si el puerto de comunicaciones es el 2 y que al mismo tiempo la tabla Async_BIOS_Port_Table tenga un valor diferente de 0, o sea que este instalado algún adaptador asíncrono y que el número de puerto de éste sea 2; de cumplirse ésta condición, se le asignará a la variable del puerto asíncrono (Async_Port) un valor de 2. En caso contrario, o sea el default (COM1), se asigna un 1 a dicha variable.

Posteriormente se extrae de la tabla correspondiente, el valor de la dirección base del dispositivo asíncrono. Inmediatamente, mediante la interrupción 14h se inicializa el UART, con los valores que se le pasaron a través de los parámetros y que fueron asignados en los correspondientes registros.

Para los siguientes pasos se deshabilitan todas las interrupciones.

A continuación lee y limpia el registro de Estado de la línea.

Después lee y limpia el registro del Buffer Receptor.

Enseguida habilita el "Input Request" del controlador 8259, para estar pendiente de cualquier petición de llamada externa.

Luego habilita la interrupción "Data Ready" del controlador 8250. O sea que esta listo para recibir datos.

Para continuar trabajando se vuelven a habilitar las interrupciones.

Una vez que se inicializó el puerto de comunicación, se indica asignándole un valor verdadero a la función.

CLOSEPORT.

Lo primero que hace este procedimiento es deshabilitar el "Input Request" del controlador 8259.

Para continuar deshabilita todas las interrupciones. Después anula las opciones asignadas al puerto serie cuando se utilizó la función OPENPORT.

Se deshabilita la interrupción "Data Ready" del controlador 8250.

Luego se habilitan las interrupciones del BIOS que fueron deshabilitadas.

Finalmente se le asigna un valor verdadero a una variable, para indicar que el puerto ha sido cerrado.

CHARREADY. Esta función es utilizada para verificar si un caracter puede ser recibido, de ser así la función regresa un valor verdadero, y si no se pueden recibir caracteres la función toma un valor falso.

SENDCH. Este procedimiento es utilizado para transmitir un caracter a través del puerto serie. El parámetro utilizado es :

C : éste es de tipo caracter, el cual es el caracter que va a transmitirse.

Para ello, la función está constantemente preguntando si la línea esta lista para transmitir caracteres, y cuando lo está deshabilita todas las interrupciones para poder trabajar, enseguida se obtiene el código ASCII del caracter a enviar, y transmite dicho código a través del puerto asíncrono. Finalmente reestablecer las interrupciones deshabilitadas.

SENDSTR. Este procedimiento se usa para transmitir una cadena de caracteres a través del puerto serie. Para tal efecto se usa :

S : éste parámetro es una cadena de caracteres, la cual va a ser transmitida.

Primero se obtiene la longitud de la cadena de caracteres que va a transmitirse, posteriormente en un ciclo que va desde uno hasta la longitud calculada para enviar cada uno de los caracteres que componen a dicha cadena, utilizando el procedimiento de SENDCH.

III.4.-REQUERIMIENTOS.

FREQUERIMIENTOS PARA ESTABLECER LA COMUNICACION EN EL SISTEMA INFORMACION OPERATIVA EN LINEA.

HARDWARE.

Para obtener los datos que se van analizar gráficamente, se tienen que captar de diferentes regiones del país, y para ello se requiere de una computadora que soporte la comunicación simultánea con diversas fuentes de información, para lo cual no está capacitada una PC; en cambio un Mainframe si puede extraer tal información.

Por lo cual se utiliza la computadora TANDEM VLX, ya que actualmente BANAMEX cuenta con este tipo de sistema con dicha estructura de comunicación.

Sin embargo un Mainframe no puede desplegar información en forma gráfica, es por ello que se requiere implementar el sistema en una Red de PCs, ya que nos da la facilidad de consultar la información simultáneamente por varios usuarios; además que una PC puede simular la terminal de una TANDEM.

A continuación mencionaremos las características básicas del hardware utilizado en el sistema ya mencionado.

CARACTERISTICAS MINIMAS DE LAS ESTACIONES DE TRABAJO (PC's.)

Procesador 80286.

12 Mhz.

Monitor VGA.

Tarjeta de red ARCNET.

CARACTERISTICAS DE LA RED LOCAL DE MICROCOMPUTADORAS.

Novell Advanced Netware 286 V2.15 Rev. C

DOS: MSDOS V4.00 on IBM_PC

Shell: V2.15 Rev. C

LAN DRIVER : Standars Microsystems ARCNET / Pure Data V1.00 IRQ
= 2, I/O Base = 2E0h, RAM Buffer at D000:0

IPX Version: 2.01

SPX Version: 1.01

Topología : Anillo Modificado.

CARACTERISTICAS DEL SISTEMA TANDEM.

TANDEM NONSTOP VLX.

SISTEMA OPERATIVO : GUARDIAN 90 B40.

INTERFAZ ELECTRICA: RS232C. X.21

6100 (CSS). Es un subsistema de comunicación de datos completa, que es usa una combinación flexible de hardware y software, diseñada para soportar varias funciones de teleproceso acompañada formalmente por varios procesos de comunicación por separado, y controladores.

- Este sistema provee flexibilidad y versatilidad, protocolos de comunicación, características en línea, e interfaces físicas que pueden ser mezcladas en muchas combinaciones.
- Diseño modular para un fácil mantenimiento.
- Soporta un amplio rango de protocolos e interfaces sobre hardware estándar.
- Diagnósticos en línea y autodiagnóstico de hardware para proveer mantenimiento y determinación de problemas.

CIU. Unidad de Interface de Comunicaciones :

- Comunicación Full Duplex.
- Conexión de bus serial y conexión para procesadores TANDEM NONSTOP a través de canales de entrada/salida.

SOFTWARE

Para desarrollar el sistema de comunicación en PC, se requiere de un lenguaje capaz de realizar una interface de comunicación constante de una PC con una TANDEM.

Es por ello que se eligió PASCAL, ya que además de las características previamente mencionadas, es un lenguaje estructurado; lo cual permite realizar rutinas que se pueden ir sumando o eliminando al sistema, para realizar pruebas y mejoras al mismo sin que se tenga que alterar todo el programa.

Y para desarrollar el sistema en dicho lenguaje, se determinó utilizar el compilador de TURBO PASCAL V5.5. A continuación se describirán las características más importantes de éste software.

Dicho paquete provee gran versatilidad en el desarrollo de sistemas, pues el editor es de fácil manejo, y cuenta con una ayuda en línea, varias opciones de compilación, depurador de programas y facilidad de integración de librerías generadas por cualquier paquete de la familia Turbo.

Otras de las cualidades con que cuenta es su gran velocidad de compilación y detección de errores, así como su gran manejo de gráficos. Además de que cuenta con programación orientada a objetos.

Para realizar la comunicación de la red local de microcomputadoras con la red TANDEM es necesario utilizar el software llamado INFORMATION XCHANGE FACILITY (IXF).

Como los programas de éste sistema tienen que acceder muchos archivos de datos, fue necesario implementarlos en un lenguaje, en el cual la manipulación de archivos fuera sencilla, y además nos permitiera implementar módulos de manera independiente, o sea que fuera estructurado. Es por ello que se eligió el lenguaje COBOL.

Para desarrollar el sistema en dicho lenguaje, se determinó utilizar el compilador COBOL 85 de TANDEM.

CAPITULO IV

INTERFACE GRAFICA AL USUARIO

Capítulo IV.

INTERFACE GRAFICA AL USUARIO

IV.1.-CONCEPTOS GENERALES DE GRAFICACION.

Las computadoras se han convertido en una herramienta poderosa para la producción rápida y económica de ilustraciones. Prácticamente no existe ninguna área en la cual no puedan utilizarse despliegues gráficos con alguna ventaja; así que no es sorprendente hallar las gráficas por computadoras en tantas aplicaciones. Aunque las primeras aplicaciones en ciencia e ingeniería tenían que basarse en equipo costoso y complicado, los adelantos en tecnología de computación han hecho de las gráficas de computadoras interactivas una herramienta práctica. Hoy día, se puede advertir que estas gráficas se utilizan rutinariamente en áreas diversas como la administración, la industria, el gobierno, arte, entretenimiento o esparcimiento, publicidad, educación, investigación, capacitación y medicina.

A menudo un programa de trazo de gráficas tendrá la capacidad de generar una variedad de tipos de gráficas, como diagramas de barras, gráficas lineales, gráficas superficiales o diagramas de pastel. Muchos programas pueden resumir datos en forma bidimensional o tridimensional. Las gráficas tridimensionales se usan comúnmente para ilustrar múltiples relaciones. En algunos casos, las gráficas tridimensionales se emplean para ofrecer una presentación más dramática o más atractiva de los datos.

Las gráficas administrativas, una de las áreas de aplicación que más rápido crece, hace un uso amplio de despliegues visuales como medio de una rápida comunicación de las vastas cantidades de información que se compilan para gerentes y otros individuos de una organización. Las gráficas y los diagramas se usan por lo general para resumir datos financieros, estadísticos, matemáticos o económicos y varias gráficas a menudo se combinan en una presentación. Sin ayuda de estas gráficas, sería difícil interpretar tablas de datos con millones de registros. En forma análoga los modelos generados por computadora se utilizan para estudiar el comportamiento de los sistemas.

Muchos sistemas de gráficas están diseñados para utilizar sistemas de coordenadas Cartesianas. Más de un sistema cartesiano puede ser referido por un programa, ya que diferentes dispositivos de salida pueden requerir diferentes sistemas de coordenadas. Además, los programas por lo general

permiten que se constituyan definiciones de imágenes en cualquier sistema de referencia cartesiano que convenga a la aplicación que se tiene a la mano. Las coordenadas referidas por un usuario se denominan **coordenadas mundiales** y las coordenadas que utiliza un dispositivo de salida particular reciben el nombre de **coordenadas de dispositivo** o bien **coordenadas de la pantalla** en caso que se trate de un monitor de video. Las definiciones de las coordenadas mundiales permiten a un usuario fijar cualquier dimensión adecuada sin verse obstaculizado por las restricciones de un dispositivo de salida determinado. Los proyectos arquitectónicos podrían especificarse en fracciones de un pie, mientras que otras aplicaciones podrían definir escalas de coordenadas en términos de milímetros, kilómetros o años-luz. Una vez dadas las definiciones de las coordenadas mundiales, el sistema de gráficas las convierte en las coordenadas de dispositivo adecuadas para su despliegue.

Un procedimiento común que se usa en los programas de gráficas consiste primero en convertir las definiciones de las coordenadas mundiales en **coordenadas de dispositivo normalizadas** antes de la conversión final a coordenadas de dispositivos específicas. Esto hace que el sistema sea lo suficientemente flexible para dar cabida a varios dispositivos de salida. Las coordenadas x y y normalizadas se transforman después en coordenadas de dispositivo (enteras) dentro de las diferencias de escalas y razones de aspecto, las coordenadas normalizadas pueden trazarse en un área cuadrada del dispositivo de salida de manera que se conserven las proporciones adecuadas. En un monitor de video, el área restante de la pantalla a menudo sirve para desplegar mensajes o listar opciones de programa intelectivos.

Las estructuras básicas de imágenes se conocen como **primitivas de salida**. Entre ellas se incluyen cadenas de caracteres y entidades geométricas, como puntos, líneas rectas, polígonos y circunferencias. Las rutinas para la generación de primitivas de salida ofrecen las herramientas básicas para la construcción de figuras.

Los **atributos** son las propiedades de las primitivas de salida. Incluyen especificaciones de color e intensidad, estilos de líneas, estilos de texto y modelos de llenado de áreas. Las funciones dentro de esta categoría pueden usarse para fijar los atributos de grupos de primitivas de salida.

Dada la definición de primitivas y atributos de una figura en coordenadas mundiales, un sistema de gráficas proyecta una vista seleccionada de la imagen en un dispositivo de salida. Las **transformaciones de vistas** se utilizan para especificar la vista que se presentará y la porción del área en el despliegue de salida que se usará.

Las imágenes pueden subdividirse en partes componentes o **segmentos**. Cada segmento define una unidad lógica de la figura. Una escena con varios objetos podría definir la construcción de cada objeto en un segmento nombrado

por separado. Las rutinas para procesar segmentos realizan operaciones como la creación, supresión y transformación de segmentos.

Por último, los sistemas de graficación contienen comúnmente varias tareas de mantenimiento, como el borrado de la pantalla de un despliegue o la inicialización de parámetros. Se pueden agrupar las funciones para realizar estos trabajos rutinarios, llamándolas operaciones de control.

El objetivo principal del software de gráficas estandarizado es la portabilidad. Cuando los paquetes se diseñan con funciones de gráficas estándar, el software puede moverse fácilmente hacia diferentes instrumentaciones y aplicaciones. Sin normas, los programas diseñados para un sistema de hardware a menudo pueden transferirse a otro sin reescribir el software.

Las organizaciones internacionales y nacionales de planeación de normas de muchos países han cooperado en un esfuerzo para crear un estándar que se acepte en general para las gráficas de computadora. Después de realizar un esfuerzo considerable, este trabajo sobre normas condujo a la creación del *sistema kernel de gráficas* (GKS). Este sistema ha sido adoptado como norma de software de gráficas por la International Standards Organization (ISO) y por varias organizaciones nacionales de normas, como el American National Standards Institute (ANSI). Aunque GKS se diseñó originalmente como paquete de gráficas bidimensional, después se creó una extensión tridimensional del GKS.

Las funciones de gráficas estándar se definen como un conjunto de especificaciones abstractas, independientes de cualquier lenguaje de programación. Para instrumentar una norma de gráficas en un lenguaje de programación determinado, debe definirse una *vinculación de lenguaje*. Esta vinculación define la sintaxis para acceder las varias funciones de gráficas que se especifican dentro de la norma. Por ejemplo, GKS especifica una función para generar una secuencia de segmentos rectilíneos conectados con el título descriptivo.

polyline (n,x,y)

Se han definido vinculaciones de lenguaje GKS para el FORTRAN, Pascal, Ada, C, PL/I y COBOL. Cada vinculación del lenguaje se define para hacer aprovechar al máximo las capacidades del lenguaje correspondiente y para manejar varios aspectos de sintaxis, como tipos de datos, paso de parámetros y errores.

Dentro de este capítulo, los algoritmos utilizados se basan en la vinculación de lenguaje GKS para Pascal.

IV.2.-DESARROLLO DE UNIDADES DE GRAFICACION.

Una consideración importante de la generación de gráficas tridimensionales es la identificación y supresión de las partes de la imagen definida que no son visibles desde una posición de observación seleccionada. Existen muchos métodos que se pueden utilizar para resolver este problema y se han creado numerosos algoritmos para eliminar las partes ocultas de los objetos en forma eficaz en diferentes tipos de aplicaciones. Algunos métodos requieren más memoria, en algunos interviene más tiempo de procesamiento y algunos sólo se aplican a tipos especiales de objetos. El método elegido para una aplicación determinada depende de factores tales como la complejidad de la escena, los tipos de objetos que se desplegarán, el equipo de disposición y si se van a generar despliegues animados o bien estáticos. En esta parte del capítulo se explicará el método que se usó para generar las gráficas tridimensionales del sistema de Información operativa en línea.

Los algoritmos de líneas y superficies ocultas a menudo se clasifican según que se refieren a definiciones de objetos en forma directa o bien con sus imágenes proyectadas. Estos dos métodos se denominan métodos de *objeto-espacio* y métodos de *imagen-espacio*, respectivamente. El primer método compara objetos y partes de objetos unos con otros para determinar qué superficies y líneas, como un todo, deben rotularse como invisibles. En un algoritmo de imagen-espacio, la visibilidad se decide por punto en cada posición de píxel sobre el plano de proyección. Muchos algoritmos de superficie oculta aplican métodos de imagen-espacio, pero los métodos de objeto-espacio pueden usarse en forma eficaz en algunos casos.

Aunque existen diferencias importantes en el método básico abordado por los diversos algoritmos de superficie y línea oculta, muchos de ellos emplean métodos de ordenamiento y coherencia para mejorar el rendimiento. El ordenamiento se utiliza para facilitar las comparaciones de profundidad mediante ordenamiento de líneas, superficies y objetos individuales de una escena de acuerdo con su distancia desde el plano de visión. Los métodos de coherencia se usan para aprovechar las regularidades de una escena. Puede esperarse que una línea de rastreo individual contenga intervalos de intensidades de píxel constantes y los modelos de líneas de rastreo a menudo cambian poco de una línea a la siguiente.

Después de hacer un análisis de los diferentes métodos de graficación, decidimos implementar la gráficas para nuestro sistema utilizando el método de ordenamiento con profundidad, el cual mencionaremos a continuación.

El método de *ordenamiento con profundidad* es una combinación de los métodos de imagen-espacio y de objeto-espacio, el cual cumple las siguientes funciones básicas:

1. Las superficies se ordenan en sentido de profundidad decreciente.
2. Las superficies se convierten con rastreo en orden, comenzando con las superficies de máxima profundidad.

Las operaciones de ordenamiento se efectúan en objeto-espacio y la conversión con rastreo de las superficies poligonales se realizan en imagen-espacio.

El método para resolver el problema de la superficie oculta algunas veces se conoce como *algoritmo del pintor*. Al crear una pintura al óleo, un artista primero pinta los colores de fondo. Después, se suman los objetos más distantes. Por último se pintan los objetos de primer plano en el lienzo sobre los objetos más distantes. Cada capa de pintura cubre la capa anterior. Mediante una técnica análoga, primero ordenamos las superficies según su distancia desde el plano de visión. Los valores de intensidad de la superficie más alejada se introducen después en el buffer de renovación. Tomando cada superficie sucesiva en turno (en orden de profundidad decreciente), "pintamos" las intensidades de la superficie que se procesaron anteriormente.

El trazo de superficies poligonales sobre el buffer de estructura según la profundidad se lleva a cabo en varias etapas. En el primer pase, las superficies se ordenan de acuerdo con el mayor valor de z en cada superficie. La superficie con la máxima profundidad (llámese S) se comparará después con las otras superficies de la lista para determinar si hay alguna superposición en la profundidad. Si no ocurre alguna superposición de profundidad, S se convierte con rastreo. Este proceso se repite después con la siguiente superficie de la lista. mientras no ocurran superposiciones, cada superficie se procesa por orden de profundidad hasta que todas se hayan convertido con rastreo. Si se detecta alguna superposición de profundidad en cualquier punto de la lista, necesitamos hacer algunas otras comparaciones para determinar si alguna de las superficies debe reordenarse.

Para cada superficie que se superpone con S , se hacen las siguientes pruebas. Si alguna de las pruebas se cumple, no se necesita reordenamiento de esa superficie. Las pruebas se enlistan en orden de dificultad creciente.

1. Los rectángulos limitantes en el plano xy de las dos superficies no se superponen.
2. La superficie S está en la "parte exterior" de la superficie de superposición.
3. La superficie de superposición está en la "parte interior" de la superficie S , relativa al plano de visión.

4. Las proyecciones de las dos superficies sobre el plano de visión no se superponen.

Tan pronto se determina que una prueba se cumple con una superficie de superposición, se sabe que esa superficie no está detrás de S . Así que pasamos a la siguiente superficie que está sobre S . Si todas las superficies de superposición pasan por lo menos una de estas pruebas, no se necesita reordenamiento y S puede convertirse con rastreo.

La prueba 1 se efectúa en dos partes. Primero se verifica si hay superposición en la dirección x , luego en la dirección y . Si alguna de estas direcciones no muestra superposición, los dos planos no pueden oscurecerse entre sí.

Podemos realizar la prueba 2 sustituyendo las coordenadas de todos los vértices de S en la ecuación del plano para la superficie de superposición y comprobando el signo del resultado. Supóngase que la superficie de superposición tiene los coeficientes del plano A', B', C' y D' . Si $A'x + B'y + C'z + D' > 0$ o para cada vértice de S , entonces la superficie S está "fuera" de la superficie de superposición.

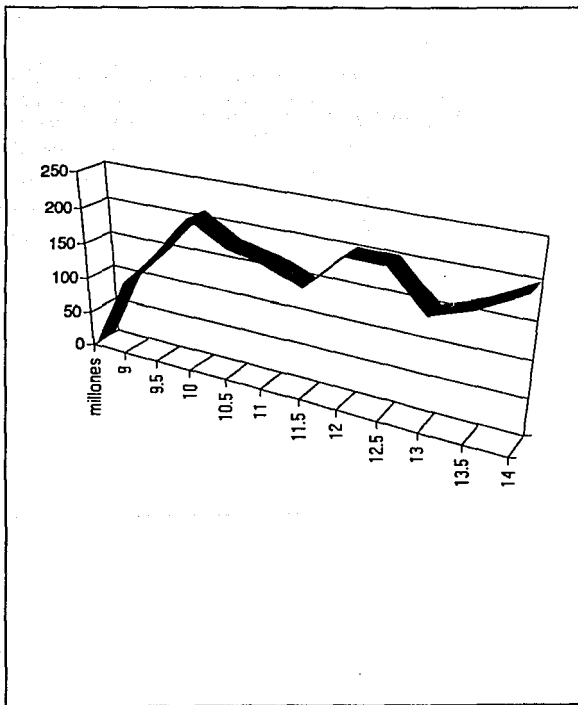
La prueba 3 se lleva a cabo con los coeficientes del plano A, B, C y D de la superficie S . Si las coordenadas de todos los vértices de la superficie de superposición satisfacen la condición $Ax + By + Cz + D < 0$, la superficie de superposición está "dentro de" la superficie S (siempre que la normal de la superficie a S apunte lejos del plano de visión).

Si las pruebas anteriores no se han cumplido, se intenta la prueba 4 verificando intersecciones entre las aristas de las dos superficies utilizando ecuaciones de rectas en el plano xy .

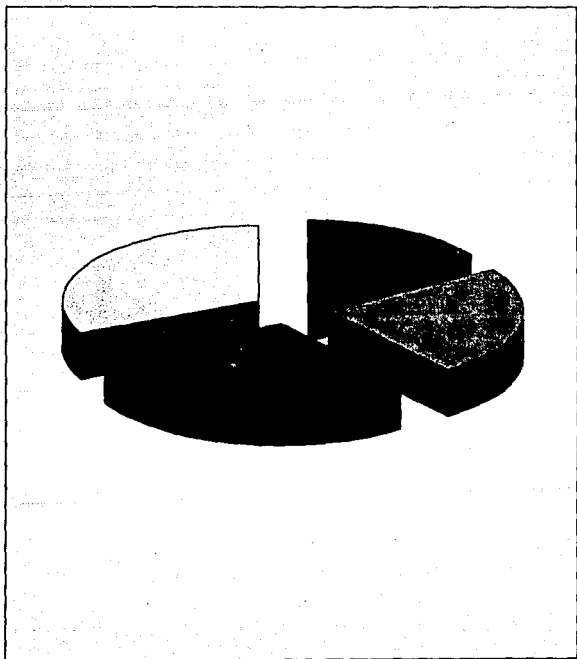
Si las cuatro pruebas llegan a fallar con una superficie de superposición S determinada, se intercambian las superficies S y S' en la lista ordenada.

Es posible que el algoritmo recién descrito entre en un ciclo infinito si dos o más superficies se oscurecen alternativamente entre sí. En tales situaciones, el algoritmo reorganizaría continuamente las posiciones de las superficies de superposición. Para evitar estos ciclos, podemos señalar alguna superficie que se haya reordenado en una posición de profundidad más lejana de manera que no pueda volver a moverse. Si se hace un intento por cambiar la superficie por segunda vez, ésta se divide en dos partes en la línea de intersección de los dos planos que se someten a comparación. La superficie original se sustituye entonces por las dos nuevas superficies y se continúa el procesamiento como antes.

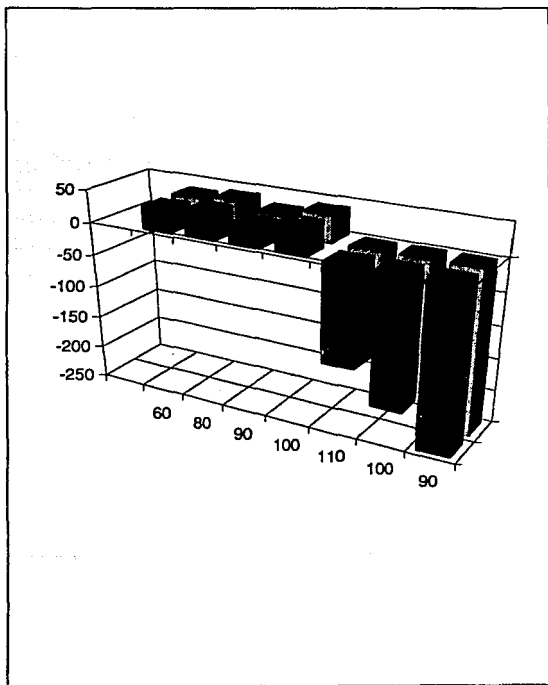
A continuación mostramos la gráfica en forma de histograma que se obtiene como resultado y que será utilizada como herramienta para que el sistema de información operativa en línea pueda desplegar su información para la toma de decisiones.



Con la gráfica de pie que se muestra a continuación se pueden representar porcentajes de cantidades numéricas, siendo así una forma más explícita de apreciar dichas cifras.



En seguida se muestra la gráfica de barras que será utilizada para desplegar los resultados de las posiciones, esta grafica tiene la característica de poder desplegar valores tanto positivos como negativos.



IV.3.-REQUERIMIENTOS.

REQUERIMIENTOS PARA REALIZAR LOS MODULOS DE GRAFICACION EN EL SISTEMA INFORMACION OPERATIVA EN LINEA.

HARDWARE.

Para implementar las gráficas que va a utilizar el sistema de información operativa en línea, es necesario contar con computadoras personales, ya que éstas cuentan con la tecnología deseada.

A continuación mencionaremos las características básicas del hardware utilizado en el sistema ya mencionado.

CARACTERISTICAS MINIMAS DE LAS COMPUTADORAS PERSONALES

Procesador 80286.

12 Mhz.

Monitor VGA.

SOFTWARE

Para implementar el módulo de graficación se requiere de un software capaz de generar gráficas tridimensionales lo suficientemente explícitas para poder tomar decisiones importantes y evitar confusiones. Tal software tiene que soportar los cambios de información que presenta el sistema de información operativa en línea, de tal forma que refresque constantemente las gráficas presentadas, para que el sistema no pierda confiabilidad. Además, tiene que ser compatible con el software desarrollado para las comunicaciones.

Después de analizar los paquetes de graficación que actualmente existen en el mercado, no encontramos un software que cumpliera con todas las características deseadas, por lo tanto llegamos a la conclusión de desarrollar nuestro propio software de graficación. Para ello requerimos de un lenguaje de programación con las cualidades mencionadas anteriormente.

Es por ello que se eligió PASCAL, ya que además de las características previamente mencionadas, es un lenguaje estructurado; lo cual permite realizar rutinas que se pueden ir sumando o eliminando al sistema, para realizar pruebas y mejoras al mismo sin que se tenga que alterar todo el programa.

Y para desarrollar el sistema en dicho lenguaje, se determinó utilizar el compilador de TURBO PASCAL V5.5. El cual cuenta con una gran facilidad para generar presentaciones gráficas con muy buena presentación y tiene la característica de la programación orientada a objetos.

CAPITULO V

IMPLEMENTACION DEL

SISTEMA DE

INFORMACION OPERATIVA

EN LINEA

Capítulo V.

IMPLEMENTACION DEL SISTEMA DE INFORMACION OPERATIVA EN LINEA

V.1.-PROGRAMACION ORIENTADA A OBJETOS

En la fase de implementación es muy importante definir las herramientas con las cuales vamos a desarrollar nuestro sistema. Finalmente determinamos utilizar como lenguaje de programación Pascal, con el apoyo de la programación orientada a objetos.

A continuación mostramos los argumentos en los que nos basamos para llegar a dicha conclusión.

Al principio de la era de la computación, la programación con lenguaje ensamblador fue la herramienta para codificar a los procesadores, manipulando los valores de los registros y las direcciones de memoria. Al mismo tiempo, el principal interés fue el compactar el código, minimizando los requerimientos de memoria y, finalmente la velocidad de ejecución. Consideraciones como el orden y la organización no fueron totalmente ignorados, pero se tomaron en un segundo término.

Después con la aparición de los lenguajes de alto nivel como C y Pascal, el objetivo de los programadores se desvió hacia procedimientos y funciones, junto con variables y tipos de datos. Al mismo tiempo, surgió el énfasis en la programación estructurada y se impuso el orden en los que a veces tenía que ser un caos.

Este fue el hecho que hizo posible que los programas se convirtieran en construcciones modulares.

Ahora con la aparición de la programación orientada a objetos (comunemente llamada OOP), la abstracción es tomada como un paso más adelante mientras que muchos detalles de operaciones de programación son ocultas casi en su totalidad a simple vista. También, con OOP, la modularidad se convirtió en la regla más que la excepción, sin embargo, como podremos apreciar, esta será una modularidad a un nuevo nivel.

Previamente, la programación orientada a objetos fue altamente restringida a aplicaciones especializadas o experimentales como la Inteligencia Artificial o los Sistemas Expertos, apareciendo en forma de un lenguaje interprete SmallTalk. El *objeto* en la programación orientada a objetos es simplemente una etiqueta genérica para un elemento de programación, pero el *objeto* en cuestión es definido, en parte, por lo que no es. Esto es porque individualmente no es ni un procedimiento, ni una función, ni una variable ni ningún tipo de dato.

Sin embargo, el *objeto* en cuestión esta compuesto de todos estos elementos y relacionado entre ellos mismos. Estos están encerrados en un nuevo tipo de modularidad, en la cual los datos, variables, procedimientos y sus relaciones están comprendidas en un nuevo elemento modular llamado objeto.

La programación orientada a objetos es una herramienta que puede ser aplicada en cualquier tarea, y no restringe su uso y pueden ser lo más general y diversa posible.

El primer ejemplo usado para demostrar los principios de la programación orientada a objetos son las aplicaciones gráficas, ya que las gráficas contienen un natural ambiente para demostrar la programación orientada a objetos, permitiendo, literalmente, ver como se van creado los objetos. Lo cual hace que para aplicaciones gráficas la programación orientada a objetos tome ventaja sobre la programación convencional.

Por lo cual este tipo de programación nos ayudará en gran medida a implementar las librerías de graficación que se utilizarán en la implementación del sistema de información operativa en línea.

El primer paso antes de comenzar con la programación fue conocer las principales propiedades de la programación orientada a objetos, las cuales son: herencia, encapsulamiento y polimorfismo, y se describirán brevemente a continuación.

Herencia.

La herencia es una propiedad de los objetos, la cual permite la creación de una jerarquía de objetos con descendientes de objetos heredando acceso al código y estructuras de datos de su ancestros.

Encapsulamiento.

Encapsulamiento es la modularidad aplicada a los datos. Es una combinación de registros con procedimientos y funciones - llamados métodos - que manipulan datos formando un nuevo tipo de datos llamado *objeto*.

En algunos lenguajes de programación orientada a objetos, el encapsulamiento se tiene que cumplir estrictamente, y los elementos de un objeto solamente pueden ser accedidos a través de métodos definidos para un objeto. Si un método no está provisto para leer o escribir elementos de datos de un objeto, entonces los datos no pueden ser accedidos directamente por el programador.

Una de las características de la programación orientada a objetos es la capacidad de crear librerías que puedan ser utilizadas por otros programas.

Polimorfismo.

Polimorfismo, es la propiedad de compartir una simple acción a través de la jerarquía de un objeto. Cada objeto en una jerarquía implementa la acción de una manera apropiada para especificar sus requerimientos.

Una vez que conocemos las ventajas que nos da la programación orientada a objetos sobre la programación convencional, podemos continuar hablando de la implementación del sistema de información operativa en línea.

V.2.-ALGORITMOS DE EJECUCION.

Una vez que analizamos la situación presentada, sintetizamos los requerimientos y las soluciones; llegamos al diseño del sistema de cómputo que va a ayudar en la toma de decisiones de los inversionistas del mercado financiero.

También determinamos la plataforma y las herramientas de desarrollo que se van a utilizar en el desarrollo del sistema.

El siguiente paso es implementar el sistema, para ello se describirán a continuación los principales módulos a codificar, y posteriormente se presentarán sus correspondientes algoritmos.

Previamente a la ejecución del sistema IOL, se tiene que correr el server question, el cual se va a encargar de comunicarse con los distintos centros de información del país para extraer la información necesaria para nuestro sistema y almacenarla en el archivo ESTINS.

Posteriormente se correrá el IOL, que está dividido en dos grandes partes.

La primera se encarga de establecer la comunicación LAN-MAINFRAME. Por el lado del mainframe se ejecutará el server llamado answer, el cual estará esperando las instrucciones por parte del programa receptor.

Por el otro lado se utilizará a una estación de trabajo de la red local como un "gateway" o computadora de comunicaciones, la cual va a correr el programa receptor, el cual configurará su puerto serial para comunicarse con el mainframe, posteriormente dialogará con el server answer, le dará las instrucciones para que realice la lectura del archivo ESTINS, y finalmente los datos que se le proporcionaron se almacenarán en la base de datos de la red local.

Tanto el server answer del mainframe, como el programa receptor de la red local, estarán corriendo durante el transcurso del día y mientras duren las actividades financieras, para mantener una constante alimentación de datos al sistema IOL.

La segunda parte del sistema es la que se encarga de manipular la información extraída de los distintos centros de información y desplegarla en forma gráfica.

Para comenzar, el sistema busca un archivo de configuración, el cual contiene la ruta de búsqueda de los archivos de datos y de las librerías de graficación.

Si no hay errores, el sistema presentará un menú principal, el cual cuenta con cuatro opciones :

1. POSICIONES. Cuando se escoge esta opción, a continuación se presenta un submenú, el cual contiene las distintas regiones que se pueden consultar para ver su posición.

Finalmente se tienen dos opciones, desplegar la posición de los instrumentos en una tabla de datos; o presentarla en forma de histogramas, ya sea en forma individual o en general todos los instrumentos y todas sus posiciones.

2. TASAS. Al elegir esta opción, es presentado un submenú, el cual contiene la lista de la regiones de los instrumentos a los cuales se le pueden consultar su correspondiente tasa.

Una vez elegida la región, se pueden elegir entre dos opciones, la primera, presentar las tasas manejadas para los instrumentos; y la otra es representar con histogramas las tasas relacionadas con cada instrumento, ya sea en forma individual o en general todas las tasas y todos los instrumentos.

3. RESULTADOS. Al seleccionar ésta parte del menú principal, es desplegada una lista de regiones, donde se pueden consultar los resultados obtenidos en la operaciones efectuadas.

Después de elegir alguna región, se puede elegir el desplegar una tabla con los resultados de las posiciones; o el mostrar dichos resultados con histogramas.

Lo mismo que en posiciones y tasas, tanto en la representación de los datos en tablas como en histogramas, los datos son refrescados constantemente para asegurar la confiabilidad del sistema.

4. SALIR. Finalmente, ésta opción es utilizada cuando se requiere salir del sistema.

Una vez que se explicó brevemente el funcionamiento del sistema se presentan los algoritmos de los procedimientos que realizan dichas tareas.

V.2.1.-ALGORITMO DE POSICIONES.

Este procedimiento obtiene de la base de datos de la red local, la información necesaria para presentar la posición de cada instrumento financiero, ya sea en una tabla de datos o mediante un histograma, según halla elegido el usuario.

Para la realización de éste procedimiento se requieren de las siguientes unidades ya implementadas :

Dos	, OpDate	, Opct	, Graficas
Instrume	, Defaults	, Cadenas	, Graph
Font	, Hor_Tiem	, Sistema	, TAccess
TAHigh	, Fechas	, Raiz	

Este procedimiento fue elaborado con la técnica de programación orientada a objetos, donde el objeto a manejar es _POSICION. Y los procedimientos asociados a tal objeto se describen a continuación.

Mayor (Arreglo arreglo de números
 N Número de elementos que tiene el arreglo)

Esta función obtiene el número mayor de un arreglo.

ALGORITMO DE EJECUCION

- 1-En una variable temporal se asigna el mínimo valor que puede tomar un número.
- 2-Se toma el primer elemento del arreglo.
- 3-Se compara el valor de la variable temporal con el del actual elemento del arreglo.
- 4-Si el valor del elemento del arreglo es mayor al de la variable temporal, la variable temporal toma dicho valor.
En caso contrario la variable temporal continúa con el mismo valor.
- 5-Se obtiene el siguiente elemento del arreglo.
- 6-Se realizan los pasos 3,4 y 5 hasta se se terminen todos los elementos del arreglo.
- 7-El valor de la variable es asignado a una variable que es la que tiene el máximo valor de dicho arreglo.

Suma (Arreglo arreglo de números
N número de elementos que tiene el arreglo)

Esta función obtiene la suma de los elementos de un arreglo.

ALGORITMO DE EJECUCION

- 1-Se le asigna un valor de cero a una variable temporal para almacenar la sumatoria.
- 2-Se posiciona en el primer elemento
- 3-A la variable temporal se le suma su valor más el valor del número posicionado.
- 4-Se mueve a la siguiente posición del arreglo.
- 5-Se realizan los pasos 3 y 4 hasta que el valor de la posición del arreglo sea igual al número de elementos (N).
- 6-El valor de la función que se regresa como resultado, es el valor final de la variable temporal.

Lectura_Posicion (No_Posiciones número de instrumentos a leer.
RCostos las posiciones de los instrumentos.
HoraStr hora de la muestra)

Esta función realiza la lectura de las posiciones de los instrumentos a desplegar. Regresa un valor verdadero si la lectura se realiza satisfactoriamente, de lo contrario regresa un valor negativo.

ALGORITMO DE EJECUCION

- 1-Se abre el archivo 'Posicion'.
- 2-Si no hubo problema en el paso anterior continuamos, sino termina el programa, indicando el correspondiente error.
- 3-Las variables que van a contener el resultado de la lectura de éste archivo se inicializan con valor cero.
- 4-Como el archivo está indexado, se realiza la búsqueda del primer registro, a través de su llave de indexación.
- 5-Se realiza la lectura del registro del cual se esta apuntando en el archivo actual.

- 6-El resultado de la lectura anterior es almacenado en una variable de paso.
- 7-Si la lectura fue bien realizada, se obtiene la hora de dicha lectura y se almacena en una variable, para que posteriormente sea desplegada.
- 8-Toda la información que fue leída anteriormente es almacenada en variables que corresponden a cada instrumento, y que posteriormente será desplegada.
- 9-Se cierra el archivo de 'POSICIONES'.

Lectura_Histograma (No_instru Número de posiciones.
Monto Cantidades a graficar.
Titulo Contiene la hora de las
 muestras a graficar.
Columna Contiene la escala de
 graficación.)

Esta función realiza la lectura de los datos de las posiciones de los instrumentos que van a ser graficados mediante histogramas. Regresa un valor verdadero si la lectura se realiza satisfactoriamente, de lo contrario regresa un valor negativo.

Calcula (RMayor Contiene la cantidad más grande.
RCosto Las posiciones de los instrumentos.
RSuma Totales de las posiciones.
No_Posiciones Número de posiciones que netran en
 el cálculo)

Este procedimiento calcula las tasas y los valores mayores de cada posición por instrumento.

ALGORITMO DE EJECUCION

- 1-Con la función mayor, obtiene por instrumento los mayores valores leídos de :
Costo.
Importe máximo.
Importe óptimo.
Importe máximo estimado.
Importe óptimo estimado.
- 2-Se calcula la tasa de cada instrumento, con un factor predefinido.

Lista_Porcentaje_Posicion (Caso 1 : Posiciones.
 2 : Porcentajes.
 X, X1, X2, Y, Y1, Y2, Y3 Parametros
 de la tabla a
 desplegar.
 RCostos Posiciones de los
 instrumentos.
 RSuma Totales.
 No_Posiciones Número de
 posiciones a
 desplegar.)

Este procedimiento despliega en una tabla cada uno de los instrumentos, y también puede desplegar la posición o el porcentaje de cada uno de ellos.

ALGORITMO DE EJECUCION

- 1.- Justificación del texto en el centro.
- 2.- Elección del tipo de texto.
- 3.- Elección del estilo de líneas delgadas.
- 4.- Cambia color en negro.
- 5.- Estilo de relleno para el despliegue de la base de la tabla.
- 6.- Despliegue de la base.
- 7.- Despliegue del borde de tabla (Rectángulo) .
- 8.- Despliegue de las líneas delgadas.
- 9.- Cambio de estilo en el tipo de línea.
- 10.- Despliegue de las líneas gruesas.
- 11.- Cálculo de la primera posición en X y Y para la primera columna y renglón a desplegar.
- 12.- Desde 1 hasta el número de plazos
 - 12.1.- Calcula la posición de los renglones.
 - 12.2.- Despliega los nombres de cada instrumento, así como de su correspondientes importes y posiciones.
 - 12.3.- Despliega la línea de cada renglón.

Lista_Captacion (RSuma	Totales de la captación
	TRP_Max	Tasa promedio de rendimiento máxima
	TRP_Opt	Tasa promedio de rendimiento óptima
	TRP_MaxC	Tasa promedio de rendimiento máxima calculada.
	TRP_OptC	Tasa promedio de rendimiento óptima calculada.
	N	Determina si la tasa es calculada o es real.
	X,X1,X2,Y,Y1,Y2,Y3	Parámetros de la tabla.)

Este procedimiento despliega en una tabla la captación estimada y la captación calculada.

ALGORITMO DE EJECUCION

- 1.- Justificación del texto en el centro.
- 2.- Elección del tipo de texto.
- 3.- Elección del estilo de líneas delgadas.
- 4.- Cambia color en negro.
- 5.- Estilo de relleno para el despliegue de la base de la tabla.
- 6.- Despliegue de la base.
- 7.- Despliegue del borde de tabla (Rectángulo) .
- 8.- Despliegue de las líneas delgadas.
- 9.- Cambio de estilo en el tipo de línea.
- 10.- Despliegue de las líneas gruesas.
- 11.- Cálculo de la primera posición en X y Y para la primera columna y renglón a desplegar.
- 12.- Desde 1 hasta el número de plazos
 - 12.1.- Calcula la posición de los renglones.
 - 12.2. Despliega los nombres de cada instrumento en la primer columna.
 - 12.3.- En la siguiente columna despliega la captacion estimada para cada instrumento.
 - 12.4.- Con una fórmula que describe el comportamiento del mercado se obtiene la captación esperada para cada instrumento, y es desplegada en la tercer columna, llamada captación calculada.
 - 12.5.- Despliega la línea de cada renglón.

Base_Tabla

ALGORITMO DE EJECUCION

- 1.- Rango para el eje X.
- 2.- Rango para el eje Y
- 3.- Define el tipo de relleno, solido y de color gris claro.
- 4.- Calculo del espacio entre cada renglón (Y1).
- 5.- Calculo del ancho de cada columna.
- 6.- Cambio de color a Negro , para las líneas
- 7.- Define tipo de texto.
- 8.- Justifica el texto hacia el centro.
- 9.- Calcula la coordenada inicial para el despliegue de los datos (Y3).
- 10.-Despliega la sombra de la tabla.
- 11.-Pone el tipo de relleno sólido .
- 12.-Traza una barra como fondo de la tabla.
- 13.-Calcula la posición en el eje y donde van los títulos de cada columna asignando este valor a la variable (Y2)
- 14.-Despliega los títulos en cada una de estas coordenadas, utilizando la unidad de font. 'Monto Costo', 'Monto Importe Maximo', 'Monto Importe Optimo', 'Resultado Estimo Optimo'.
- 15.-Define el tipo de texto, justificando el texto en el centro .
- 16.-Despliega el título de cada renglón que corresponde a los plazos
- 17.-Ejecuta las rutinas de tiempo y fechas de la unidad de tiempo.
- 18.-Despliega la Hora y la Fecha actual.

Refresca_Importe_Porcentaje.

Monitorea constantemente las posiciones de los instrumentos.

ALGORITMO DE EJECUCION

- 1.-Ejecuta el procedimiento Base_Tabla.
- 2.-La hora actual pasa a ser la hora anterior.
- 3.-Repite hasta que se presione alguna tecla :
 - 3.1.-Actualiza la posición.
 - 3.2.-Despliega las posiciones actualizadas.

Histograma.

Dibuja el histograma del comportamiento de la posición de los instrumentos conforme pasa el tiempo.

ALGORITMO DE EJECUCION

- 1.-Se determinan las coordenadas de la ventana donde va a desplegarse el histograma.
- 2.-Se dibuja el histograma.
- 3.-Se define el color de las letras de los mensajes.
- 4.-Se despliegan los títulos de los histogramas.

Base_Tabla_Captacion.

Forma la base para la tabla de captación estimada y obtiene sus parámetros.

ALGORITMO DE EJECUCION

- 1.- Rango para el eje X.
- 2.- Rango para el eje Y
- 3.- Define el tipo de relleno, sólido y de color gris claro.
- 4.- Cálculo del espacio entre cada renglón (Y1).
- 5.- Cálculo del ancho de cada columna.
- 6.- Cambio de color a Negro, para las líneas
- 7.- Define tipo de texto.
- 8.- Justifica el texto hacia el centro.
- 9.- Calcula la coordenada inicial para el despliegue de los datos (Y3).
- 10.-Despliega la sombra de la tabla.
- 11.-Pone el tipo de relleno sólido.
- 12.-Traza una barra como fondo de la tabla.
- 13.-Calcula la posición en el eje y donde van los títulos de cada columna asignando este valor a la variable (Y2)
- 14.-Despliega los títulos en cada una de estas coordenadas, utilizando la unidad de font. 'Monto Costo', 'Monto Importe Maximo', 'Monto Importe Optimo', 'Resultado Estimo Optimo'.
- 15.-Define el tipo de texto, justificando el texto en el centro.
- 16.-Despliega el título de cada renglón que corresponde a los plazos
- 17.-Ejecuta las rutinas de tiempo y fechas de la unidad de tiempo.
- 18.-Despliega la Hora y la Fecha actual.

Refresca_Captacion

Monitorea la captación de la posición.

ALGORITMO DE EJECUCION

- 1.-Ejecuta el procedimiento Base_Tabla_Captación.
- 2.-La hora actual pasa a ser la hora anterior.
- 3.-Repite hasta que se presione alguna tecla :
 - 3.1.-Obtiene la diferencia de tiempo entre la hora actual y la de la última muestra.
 - 3.2.-Obtiene la captación estimada.
 - 3.3.-Despliega la captación estimada.
 - 3.4.-Detecta si una tecla fue presionada.

Refresca_Histograma

ALGORITMO DE EJECUCION

- 1.-Despliega el titulo del correspondiente histograma .
- 2.-La hora actual pasa a ser la hora anterior.
- 3.-Repite hasta que una tecla sea presionada :
 - 3.1.-Despliega el histograma actualizado de las posición de los instrumentos.
 - 3.2.-Detecta si una tecla fue presionada.

Constructor_Init

Inicializa y obtiene memoria para el objeto
Obtiene los valores de los parámetros siguientes :
Código de la región
Código de Opción (Tabla de Posiciones o Histograma)
Tipo de Histograma
Clave del instrumento
Parámetros

ALGORITMO DE EJECUCION

- 1.- Ejecuta Init de la unidad ROOT.
- 2.- Obtiene parámetros.
- 3.- Ejecuta Pantallal de la unidad instrume.

Destructor_Init

Libera la memoria reservada por INIT

ALGORITMO DE EJECUCION

- 1.- Ejecuta Done de la Unidad Root

Run

Ejecuta el objeto de acuerdo a los parámetros obtenidos en INIT.

ALGORITMO DE EJECUCION

- 1.- Dependiendo de la Opción seleccionada ejecuta Monitorea_Posicion o Monitorea_Histograma.

Programa Principal

ALGORITMO DE EJECUCION

- 1.- Reserva Memoria
- 2.- Ejecuta el objeto (Run)
- 3.- Libera Memoria

V.2.2.-DESARROLLO DEL PROGRAMA DE TASAS

Este sistema nos permite consultar una serie de histogramas, y tablas referentes a las Tasas de rendimiento Óptima y Máxima, para las operaciones de Cash y Spot evaluando la cantidad para los diferentes plazos en cada Instrumento del Mercado de Dinero a nivel de Región.

UNIDADES USADAS

Dos , OPCrt , OPDate , Hor_Tiem, Instrume,
Cadenas, Defaults, Gráficas, Font , Graph ,
TAccess, TAHigh , Fechas , Sistema, Raiz ;

DEFINICION DEL OBJETO TASAS

```
Function Lectura_Datos(   Var _TR   : TR;  
                        Var Monto  : Montos;  
                        Var Titulo : Mercado )  
  
Procedure mascara;  
Procedure Lista_Tasas( X, X1, X2,  
                      Y, Y1, Y2, Y3 : Integer  
                      _TR           : TR );  
Procedure Base_Tabla( Var X, X1, X2,  
                     Y, Y1, Y2, Y3 : Integer );  
Function Tabla_Tasas( X , X1, X2,  
                    Y , Y1, Y2, Y3      : Integer )  
Procedure Mascara_Histograma( Titulo      : String );  
Procedure Monitorea_Tasas;  
Procedure Histograma(Var Datos, Titulos, N : Integer);  
Procedure Monitorea_Histograma;  
Constructor Init(      Config           : Configuracion  
);  
Procedure Run;  
Destructor Done;
```

```
=====
_Tasas.Mascara :
  Despliega los títulos para las tablas de Tasas.
=====
```

ALGORITMO DE EJECUCION

- 1.- Fondo en Color Gris con marco en blanco.
- 2.- Justifica el Texto al centro
- 3.- Llama al procedimiento de FONT33D de la unidad de FONT para desplegar las cadenas siguientes
 'Tasas por Instrumento'
 Nombre de la Región
 Nombre del instrumento seleccionado
- 4.- Despliega el título de las operaciones de Cash y Spot que serán las mostradas.
- 5.- Cambia tipo de texto.
- 6.- Ejecuta Captura_Boton para desplegar el comando habilitado mediante la tecla de función : ESC Regresar.

```
=====
_Tasas.Lectura_Datos :
  _TR      Registro de tasas de rendimiento
  Monto    Contiene información de todas las tasas de
           rendimiento
  Títulos Plazos de inversión
  Regresa  True  si la lectura fue satisfactoria
           False en Caso de Existir problemas con los
           datos
=====
```

ALGORITMO DE EJECUCION

- 1.- Inicializa variables
- 2.- Utiliza la unidad de raiz , ejecutando la rutina de Abre_Archivo.
- 3.- Utiliza la unidad raiz, para obtener la llave de la región ejecutando la función ObtenLlaveNodo.
- 4.- Realiza la consulta de los instrumentos y plazos de vencimiento existentes para la región.


```

=====
_Tasas.Lista_Tasas :
  Lista las tasas en la tabla
  X, X1, X2,      Parametros de la tabla en X
  Y, Y1, Y2, Y3 : Parametros de la tabla en Y
  _TR           : Tasas de Rendimiento
=====

```

ALGORITMO DE EJECUCION

- 1.- Justificación del texto en el centro
- 2.- Elección del tipo de texto
- 3.- Elección del estilo de líneas delgadas
- 4.- Cambia color en negro
- 5.- Estilo de relleno para el despliegue de la base de la tabla.
- 6.- Despliegue de la base.
- 7.- Despliegue del borde de tabla (Rectángulo) .
- 8.- Despliegue de las líneas delgadas.
- 9.- Cambio de estilo en el tipo de línea
- 10.- Despliegue de las líneas gruesas.
- 11.- Calculo de la primera posición en X y Y para la primera columna y renglón a desplegar.
- 12.- Desde 1 hasta el numero de plazos
 - 12.1.- Calcula la posición de los renglones.
 - 12.2.- Despliega los valores de los plazos de vencimiento y cada uno de sus montos para las diferentes tasas.
 - 12.3.- Despliega la línea de cada renglón.

```

=====
=_Tasas.Base_Tabla :
  Obtiene los parámetros de la tabla y despliega los
  títulos de misma.
  Parámetros
  X, X1, X2, Y, Y1, Y2, Y3
=====

```

ALGORITMO DE EJECUCION

- 1.- Rango para el eje X.
- 2.- Rango para el eje Y
- 3.- Define el tipo de relleno, solido y de color gris claro.
- 4.- Calculo del espacio entre cada renglón (Y1).
- 5.- Calculo del ancho de cada columna.
- 6.- Cambio de color a Negro , para las líneas
- 7.- Define tipo de texto.
- 8.- Justifica el texto hacia el centro.
- 9.- Calcula la coordenada inicial para el despliegue de los datos (Y3).
- 10.-Despliega la sombra de la tabla.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

- 11.-Pone el tipo de relleno sólido .
- 12.-Traza una barra como fondo de la tabla.
- 13.-Calcula la posición en el eje y donde van los títulos de cada columna asignando este valor a la variable (Y2)
- 14.-Despliega los títulos en cada una de estas coordenadas, utilizando la unidad de font. 'TR / MAX' , 'TR / OPT' , 'TR / MAX', 'TR / OPT', 'PLAZO', '(días)'
- 15.-Define el tipo de texto, justificando el texto en el centro .
- 16.-Despliega el título de cada renglón que corresponde a los plazos
- 17.-Ejecuta las rutinas de tiempo y fechas de la unidad de tiempo.
- 18.-Despliega la Hora y la Fecha actual.

```
=====
_Tasas.Tabla_Tasas :
  X, X1, X2, Y, Y1, Y2, Y3   Parámetros de la tabla
  Regresa True : Si los datos se leen correctamente
                False : Problemas al leer Datos
=====
```

ALGORITMO DE EJECUCION

- 1.-Inicializa las banderas.
- 2.-Ejecuta Lectura_Datos validando su resultado
- 3.-Si no se ejecuto la lectura correctamente se asigna a la bandera el valor de Falsa y se sale de este procedimiento.
- 4.-Si fue correcta la lectura se ejecuta la rutina de Lista_Tasas pasando los parámetros correspondientes de las posiciones a desplegar.

```
=====
_Tasas.Monitora_Tasas :
  Actualiza continuamente los valores de la tabla de
  Tasas
=====
```

ALGORITMO DE EJECUCION

- 1.- Ejecuta Base_Tabla
- 2.- Ejecuta Tabla_Tasas validando su resultado :
True : continua Ejecución
False : Sale de procedimiento
- 3.- Obtiene el tiempo
- 4.- Continúa la ejecución del procedimiento,
verificando el tiempo de muestreo.
- 5.- Permanece en el loop hasta pulsar el comando de salida(ESC).

```

=====
_Tasas.Traza_Histograma :
  Titulo   : Descripción de la tasa de rendimiento a
            gráficar
  Datos    : Información a gráficar
  Títulos  : Plazos de Inversión
  N        : Determina del Número de histogramas
=====

```

ALGORITMO DE EJECUCION

- 1.- Verifica si se desea ver todos los histogramas de las tasas de rendimiento o sólo los histogramas individuales.
 - 1.1 Si se desean ver todos los histogramas, despliega el nombre de cada tasa de rendimiento empleando FONT3D de acuerdo al color que le corresponde en la gráfica.
 - 1.2 Continúa la ejecución
- 2.- Obtiene coordenadas para el recuadro en el cual se dibujará la gráfica
- 3.- Ejecuta el procedimiento Histogramas.

```

=====
_Tasas.Mascara_Histograma :
  Dibuja los Títulos para las gráficas de histograma
=====

```

ALGORITMO DE EJECUCION

- 1.- Fondo en Color Gris con marco en blanco.
- 2.- Dibuja sombras
- 3.- Obtiene y despliega la hora y la fecha.
- 4.- Define el tipo de texto
- 5.- Llama al procedimiento de FONT33D de la unidad de FONT para desplegar la las cadenas siguientes
 - Descripción del instrumento seleccionado
 - Descripción de la tasa seleccionada
- 6.- Cambia tipo de texto.
- 7.- Ejecuta Captura_Boton para desplegar el comando habilitado mediante la tecla de función : ESC Regresar.

```
=====
_Tasas.Monitorea_Histograma :
  Actualiza la información del histograma cada TM unidades
  de tiempo.
=====
```

ALGORITMO DE EJECUCION

- 1.- Ejecuta Lectura_Datos Validando sus resultado:
True : Continua
False : Termina Procedimiento
- 2.- Ejecuta Mascara_Histograma dependiendo de la
tasa de rendimiento seleccionada.
- 3.- Ejecuta Histograma de acuerdo al parámetro
anterior.
- 4.- Obtiene hora actual.
- 5.- Dentro un loop realiza:
 - 5.1 Muestreo de las tasas de rendimiento
cada TM unidades de tiempo.
 - 5.2 Actualiza la hora en pantalla mediante
Verifica_Tiempo.
 - 5.3 Verifica si el comando pulsado es ESC.

```
=====
_Tasas.Init :
  Inicializa y obtiene memoria para el objeto
  Obtiene los valores de los parámetros siguientes :
  Código de la región
  Código de Opción ( Tabla de Tasas o Histograma )
  Tipo de Histograma
  Clave del instrumento
  Parámetros
  Procedimiento de configuración ( Procedure Configura
  de la unidad Config.)
=====
```

ALGORITMO DE EJECUCION

- 1.- Ejecuta Init de la unidad ROOT
- 2.- Obtiene parámetros
- 3.- Ejecuta Pantalla1 de la unidad instrume

```
=====
_Tasas.Run :
Ejecuta el objeto de acuerdo a los parámetros
obtenidos en INIT
=====
```

ALGORITMO DE EJECUCION

- 1.- Dependiendo de la Opción seleccionada ejecuta Monitorea_Tasas o Monitorea_Histograma.

```
=====
_Tasas.Done :
Libera la memoria reservada por INIT
=====
```

ALGORITMO DE EJECUCION

- 1.- Ejecuta Done de la Unidad Root

```
=====
Programa Principal
=====
```

Variables

T : Apuntador al Objeto _Tasas

ALGORITMO DE EJECUCION

- 1.- Reserva Memoria
- 2.- Ejecuta el objeto (Run)
- 3.- Libera Memoria

V.2.3.-DESARROLLO DEL PROGRAMA DE RESULTADOS

Este sistema realiza la explotación de información para obtener , los resultados en pérdidas y ganancias de la colocación de la posición desplegando en gráficas , los porcentajes y montos obtenidos a nivel regional para una buena percepción en la toma de decisiones.

UNIDADES USADAS

```
Dos      ,Opdate  ,Opct   ,Gráficas ,
Instrume ,Defaults ,Cadenas ,Font   ,
Sistema  ,Hor_Tiem ,TAccess ,TAHigh ,
Raiz     ,Graph   ,UPastel ,OpKey  ,
```

DEFINICION DEL OBJETO RESULTA

```
Function Menu : Boolean; virtual;
Procedure Botones;
Procedure Mascara( Var X, X1, X2, Y, Y1, Y2,
Y3:Integer);
Procedure Asigna_Porcentajes;
Procedure Despliega_Resultados( X, X1, X2, Y1, Y, Y2,
Y3:Integer);
Function Lectura_Datos : Boolean;
Procedure Nombres ( X, Y: Integer; P: Cantidades;
N:integer);
Procedure Despliega_Pastel;
Procedure Monitorea_Resultados;
Procedure Mascara_Pastel;
Procedure Monitorea_Pastel;
Procedure Grafica_Barras;
Procedure Mascara_Barras;
Procedure Monitorea_Barras;
Procedure Run;
Constructor Init;
Destructor Done;
```

```
=====
_Resultados.Menu :
  Verifica si el comando habilitado fue la tecla F1 ó
  ESC
=====
```

ALGORITMO DE EJECUCION

- 1.- Inicializa la bandera
- 2.- Indentifica el Código de la llave
 - 2.1.- Si es F1 ejecuta la función Ayuda1 de la unidad de INSTRUME. Posteriormente ejecuta Limpia_Buffer.
 - 2.2.- Si es ESC , cambia el valor de la bandera.
 - 2.3.- Si es cualquier otra tecla ejecuta Sonido.

```
=====
_Resultados.Botones :
  Despliega los botones con las diferentes funciones
  a seleccionar en la pantalla.
=====
```

ALGORITMO DE EJECUCION

- 1.- Calculo de las coordenadas Máximas para el eje X y Y.
- 2.- Define el tipo de Texto.
- 3.- Ejecuta Captura_Boton para desplegar los comandos habilitados mediante las teclas de función :
F1 Ayuda
Esc Terminar.
- 4.- Ejecuta las rutinas de tiempo y fecha de la unidad de tiempo.
- 5.- Cambia el tipo de Texto, justificando el texto al Centro.
- 6.- Cambia el color a amarillo.
- 7.- Despliega la Hora y la Fecha actual.

```
=====
_Resultados.Mascara :
  Despliega los títulos para la tabla de Resultados.
=====
```

ALGORITMO DE EJECUCION

- 1.- Fondo en Color Gris con marco en Azul Claro.
- 2.- Define el tipo de Texto
- 3.- Justifica el Texto al Centro
- 4.- Llama al procedimiento de FONT33D de la unidad de FONT para desplegar las siguientes cadenas
'Resultados'
Nombre de la Región
- 5.- Define las coordenadas Máximas en el eje X y Y
- 6.- Cálculo de la primera posición en X y Y para la primera columna y renglón a desplegar.
- 7.- Elección del estilo de líneas delgadas
- 8.- Cambia color en Negro
- 9.- Ejecuta la Rutina de Sombra.
- 10.- Define el tipo de relleno para la Base de la Tabla.
- 11.- Despliega la base de la Tabla
- 12.- Define el tipo de Texto justificando a la Izquierda.
- 13.- Ejecuta la rutina de FONT33D , desplegando las siguientes cadenas:
'INSTRUM'
'IMPORTE'
'PORCENTAJE'.

```
=====
_Resultados.Asigna_Porcetajes :
  Calcula el Porcentaje sobre el importe de Cada Instrumento.
=====
```

ALGORITMO DE EJECUCION

- 1.- Realiza una suma del importe de cada instrumento obteniendo un Importe Total.
- 2.- Calcula el porcentaje de cada instrumento con respecto al Importe Total.


```

=====
_Resultados.Despliega_Resultados :
  Despliega los Resultados para cada instrumento ,
  dibujando las lineas de la Tabla.
  Parámetros :
    X, X1, X2, Y1, Y, Y2, Y3 - Coordenadas de las
    lineas para desplegar la división entre renglones.
=====

```

ALGORITMO DE EJECUCION

- 1.- Cambia el color a Negro.
- 2.- Define el estilo de lineas delgadas.
- 3.- Define el estilo de llenado para el fondo de la tabla.
- 4.- Despliega la base de la tabla, trazando una barra.
- 5.- Despliegue de las lineas verticales
- 6.- Cambio del tipo de Texto
- 7.- Desde 1 hasta el numero de Instrumentos
 - 7.1.- Justifica el Texto a la Izquierda.
 - 7.2.- Calcula la posición del renglón
 - 7.3.- Despliega los nombres de los instrumentos.
 - 7.4.- Justifica el Texto a la derecha.
 - 7.5.- Despliega el importe del instrumento, y porcentaje
 - 7.6.- Dibuja la línea horizontal.

```

=====
Function _Resultados.Lectura_Datos :
  Regresa True si la Lectura fue satisfactoria
  False en caso de Existir problemas con los
  datos
=====

```

ALGORITMO DE EJECUCION

- 1.- Inicializa variables
- 2.- Utiliza la unidad de raiz , ejecutando la rutina de Abre_Archivo.
- 3.- Utiliza la unidad raiz, para obtener la llave de la región ejecutando la función ObtenLlaveNodo.
- 4.- Realiza la consulta del importe de cada instrumento existente para la región.
- 5.- Ejecuta Asigna_Porcentajes.

```
=====
_Resultados.Nombres X, Y : Coordenadas para los títulos
                    P   : Porcentajes a graficar
                    N   : Numero de Instrumentos.
=====
```

ALGORITMO DE EJECUCION

- 1.- Calcula el ancho de las barras a partir del parámetro en X.
- 2.- Justifica el texto a la izquierda
- 3.- Define el estilo del texto
- 4.- Desde 1 hasta el Numero de Instrumentos
 - 4.1 - Calcula las coordenadas en Y para dibujar las barras.
 - 4.2 - Dibuja la barra tomando la coordenada antes calculada.
 - 4.3 - Dibuja la sombra de la barra.
 - 4.4 - Ejecuta la rutina FONT3D desplegando el nombre del instrumento.
 - 4.5 - Justifica el texto a la derecha.
 - 4.6 - Ejecuta FONT3D para desplegar el porcentaje correspondiente al instrumento.

```
=====
_Resultados.Despliega_Pastel
```

Despliega el Pastel representando rebanadas para los porcentajes de cada instrumento.

```
=====
ALGORITMO DE EJECUCION
```

- 1.- Ejecuta la rutina de Nombres.
- 2.- Define el tipo de relleno.
- 3.- Dibuja una Barra como fondo del pie.
- 4.- Ejecuta la rutina TRAZA_PASTEL de la unidad de UPASTEL.

=====

Resultados.Monitorrea_Resultados

Actualiza continuamente los valores de la tabla de Resultados

=====

ALGORITMO DE EJECUCION

- 1.- Ejecuta Lectura_Datos validando su resultado:
TRUE : Continua
FALSE : Termina Procedimiento
- 2.- Ejecuta Mascara.
- 3.- Obtiene hora actual.
- 4.- Ejecuta Despliega_Resultados.
- 5.- Justifica el Texto al centro.
- 6.- Define el estilo del Texto.
- 7.- Mientras sea TRUE :
 - 7.1 - Obtengo la hora actual , verificando el tiempo de muestreo
 - 7.2 - Ejecuta Despliega_Resultados.
 - 7.3 - Justifica el texto al centro.
 - 7.4 - Define el estilo del Texto.
- 8.- Verifica el tiempo mostrandolo a la pantalla.
- 9.- Permanece en el LOOP hasta pulsar el comando de salida.

=====

Resultados.Mascara_Pastel :

Despliega los títulos de la pantalla, para la tabla de Resultados.

=====

ALGORITMO DE EJECUCION

- 1.- Define el estilo de Texto
- 2.- Ejecuta la rutina de Fondo de color gris.
- 3.- Define el tipo de relleno solido.
- 4.- Dibuja Barras para el fondo.
- 5.- Justifica el Texto en el Centro.
- 6.- Ejecuta la rutina de FONTS desplegando los siguientes títulos:
'SISTEMA DE INFORMACION OPERATIVA'.
'RESULTADOS EN REGION(CÓDIGO)'.
7.- Dibuja una sombra para la tabla.

- 8.- Define el estilo de linea con el color Negro.
- 9.- Dibuja un rectángulo
- 10.- Ejecuta la rutina de botones.

=====

Resultados.Monitoria_Pastel :

Actualiza la informacion del Pastel , cada TM unidades de tiempo.

=====

ALGORITMO DE EJECUCION

- 1.- Ejecuta Lectura_Datos validando su resultado:
TRUE : Continua
FALSE : Termina Procedimiento
- 2.- Ejecuta Mascara_Pastel.
- 3.- Obtiene hora actual.
- 4.- Ejecuta Despliega_Pastel.
- 5.- Justifica el Texto al centro.
- 6.- Define el estilo del Texto.
- 7.- Mientras sea TRUE :
7.1 - Obtengo la hora actual , verificando el tiempo de muestreo
7.2 - Ejecuta Despliega_Pastel.
7.3 - Justifica el texto al centro.
7.4 - Define el estilo del Texto.
- 8.- Verifica el tiempo mostrandolo a la pantalla.
- 9.- Permanece en el LOOP hasta pulsar el comando de salida.

=====

Resultados.Grafica_Barras :

I : Contador para el Color de la Barra
Color : Colores de las Barras;
Ventana : Ventana para la gráfica.

=====

ALGORITMO DE EJECUCION

- 1.- Desde 1 hasta el numero de Instrumentos.
1.1.- Asigno los colores para cada barra.
- 2.- Define el tamaño de la ventana.
- 3.- Ejecuta la rutina de VERTICALES de la unidad de Gráficas.

=====
_Resultados.Mascara_Barras :

Dibuja los títulos para las gráficas de barras.

=====
ALGORITMO DE EJECUCION

- 1.- Fondo en Color Gris.
- 2.- Define tipo de relleno Negro
- 3.- Dibuja una Barra para el fondo de la gráfica.
- 4.- Define el estilo del Texto.
- 5.- Justifica el Texto en el Centro.
- 6.- Ejecuta la rutina de FONT33D desplegando el siguiente titulo 'RESULTADOS'.
- 7.- Ejecuta la rutina de Botones.

=====
_Resultados.Monitorea_Barras :

Actualiza la informacion del la gráfica de barras
cada TM unidades de tiempo.

=====
ALGORITMO DE EJECUCION

- 1.- Ejecuta Lectura_Datos validando su resultado:
TRUE : Continua
FALSE : Termina Procedimiento
- 2.- Ejecuta Mascara_Barras.
- 3.- Ejecuta Grafica_Barras.
- 4.- Obtiene hora actual.
- 5.- Justifica el Texto al centro.
- 6.- Define el estilo del Texto.
- 7.- Mientras sea TRUE :
 - 7.1 - Obtengo la hora actual , verificando el tiempo de muestreo
 - 7.2 - Ejecuta Grafica_Barras.
 - 7.3 - Justifica el texto al centro.
 - 7.4 - Define el estilo del Texto.
- 8.- Verifica el tiempo mostrandolo a la pantalla.
- 9.- Permanece en el LOOP hasta pulsar el comando de salida.

=====

_Resultados.Run :

Selecciona la Opción deseada por el Menu.

=====

ALGORITMO DE EJECUCION

- 1.- Verifica la opción y ejecuta según el caso:
Monitorea_Resultados.
Monitorea_Barras.
Monitorea_Pastel.

=====

_Resultados.Init :

Inicializa y obtiene memoria para el objeto
Obtiene los valores de los Parámetros siguientes
Obtiene el numero de Instrumentos
Clave del Instrumento
Código de la Región
Código de la Opción
Parámetros
Procedimiento de configuración (Procedure
Configura de la unidad de Config)

=====

ALGORITMO DE EJECUCION

- 1.- Ejecuta init de la unidad de ROOT
- 2.- Ejecuta Pantalla de la unidad de INSTRUME
- 3.- Obtiene Parámetros.

=====

_Resultados.Done :

Libera la memoria reservada por INIT.

=====

ALGORITMO DE EJECUCION

- 1.- Ejecuta Done de la unidad de ROOT.

=====

PROGRAMA PRINCIPAL

=====

Variables

R : Apuntador al Objeto_Resultados

ALGORITMO DE EJECUCION

- 1.- Reserva Memoria
- 2.- Ejecuta el objeto (Run)
- 3.- Libera Memoria.

V.3.-PRESENTACION.

Es muy importante que todo sistema desarrollado tenga una buena presentación y sea fácil de utilizar, esto es, que el sistema para que sea ejecutado no necesite que el usuario le proporcione muchos parámetros, convirtiéndose así en un sistema complicado y difícil de manejar.

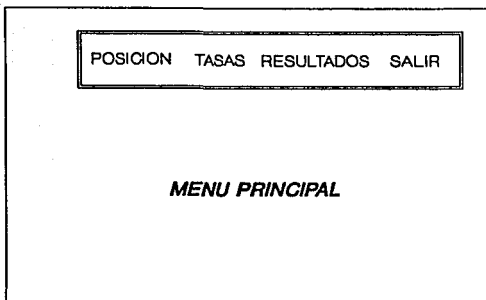
Para que el sistema de información operativa en línea tenga una interface agradable al usuario, se creará una serie de menús que le ayudarán al usuario a navegar a través del sistema, llevándolo a la opción que el requiera sin tener que teclear parámetros, solamente eligiendo una de las opciones que se listan.

V.3.1.-DESARROLLO DEL MENU.

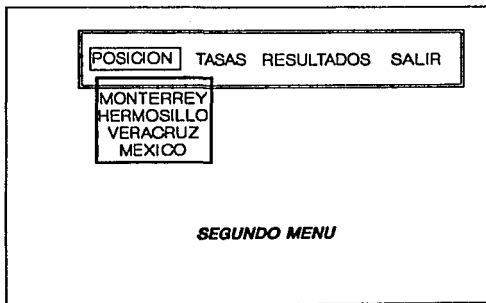
Para desarrollar la interface que se utilizará para que el usuario se comunique con el sistema, primero es necesario determinar cual es la mejor interface, esto es, la que sea más cómoda para el usuario, que no le lleve mucho tiempo en aprender a utilizarla, que sea dinámica, fácil de implementar y dar mantenimiento.

Lo anterior nos llevó a evaluar entre las diferentes técnicas de presentación que existen para los sistemas. Inclinandonos al desarrollo de menús de persiana, los cuales funcionan de la siguiente forma:

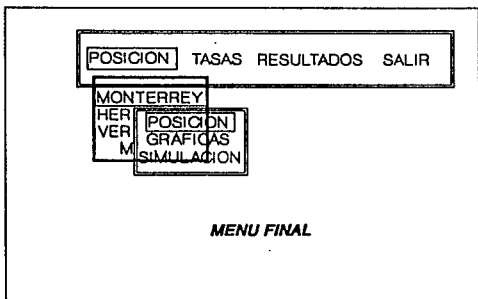
La primer presentación, es una lista horizontal conteniendo los principales módulos con que cuenta el sistema, tal y como se muestra a continuación.



La siguiente presentación aparece cuando se elige a través de las flechas del teclado la opción deseada y se presiona <enter>. Entonces, la opción elegida es remarcada y del letrero de dicha opción se desprende como si fuese una persiana una lista vertical conteniendo las opciones disponibles del módulo elegido. Tal y como ve puede observar a continuación.



El procedimiento anterior se sigue en cada opción de la lista hasta llegar a la última opción que será la que ejecute el correspondiente procedimiento, permitiendo al usuario tener a la vista toda la serie de opciones que eligió para poder ejecutar el sistema.



Para regresar al menú anterior se presiona la tecla de escape, esto es posible hasta llegar al menú principal.

Finalmente para salir se elige la opción <salir>, y automáticamente el usuario saldrá de la aplicación.

El procedimiento anterior es muy sencillo de llevar a cabo, así el usuario puede ejecutar el sistema sin temor de equivocarse o de no pasar los parámetros correctos.

Por otra parte, este tipo de menús es fácil de implementar, y para ello utilizaremos OBJECT PROFESIONAL, que es una herramienta que nos ayuda a generar los menús necesarios para que el sistema funcione. Dicha herramienta genera el correspondiente código en lenguaje *pascal*, el cual es nuestro lenguaje de programación.

A continuación se describen los algoritmos utilizados para generar los menús del sistema de Información operativa en línea.

V.3.2.-ALGORITMO DEL MENU.

Unidades

```
Defaults, OpPick , OpDos , Dos , Instrume,  
OpInline, OpString, OpRoot , OpCrt , OpColor ,  
OpFrame , OpCmd , OpWindow, OpMenu, OPExec ;
```

Tipos

```
Instrumento = Record  
  Clave : String[4];  
  Abreviatura : String[10];  
  Descripcion : String[40];  
End;
```

Constantes

```
NI = 13; { Numero de instrumentos del mercado de Dinero }  
{ Instrumentos del Mercado de Dinero }  
Instrumentos : Array [1..NI] Of Instrumento =  
{ Clave : '301'; Abreviatura: 'CETE' ;  
  Descripcion : 'Certificados de la Tesorería' },  
{ Clave : '302'; Abreviatura: 'PACO' ;  
  Descripcion : 'Papel Comercial' },  
{ Clave : '303'; Abreviatura: 'PAEX' ;  
  Descripcion : 'Papel Extra Bursatil' },  
{ Clave : '304'; Abreviatura: 'ABS' ;  
  Descripcion : 'Aceptaciones Bancarias' },  
{ Clave : '305'; Abreviatura: 'ABSP' ;  
  Descripcion : 'Aceptaciones Bancarias Privadas' },  
{ Clave : '306'; Abreviatura: 'TESO' ;  
  Descripcion : 'Bonos de la Tesorería' },  
{ Clave : '307'; Abreviatura: 'PAQ' ;  
  Descripcion : 'Paquetes de Inversión' },  
{ Clave : '308'; Abreviatura: 'PAGA' ;  
  Descripcion : 'Pagarés de la Federación' },  
{ Clave : '309'; Abreviatura: 'BOND' ;  
  Descripcion : 'Bonos de Desarrollo' },  
{ Clave : '310'; Abreviatura: 'CEPL' ;  
  Descripcion : 'Certificados de Plata' },  
{ Clave : '311'; Abreviatura: 'FACT' ;  
  Descripcion : 'Factoraje BANAMEX' },  
{ Clave : '312'; Abreviatura: 'AJUS' ;  
  Descripcion : 'Ajusta Bonos' },  
{ Clave : '313'; Abreviatura: 'BODI' ;  
  Descripcion : 'Bondis' };
```

```
=====
Procedure LlaveChoice
Manda las llaves para cada una de las opciones de lista
existente para los instrumentos del Mercado de Dinero.
Item Numero de opción solicitada
Mode Bandera de selección
IString Cadena para regresar la opción
PickPtr Apuntador a lista de opciones en video
=====
```

```
=====
Function GetInstrumento
Esta función despliega una lista de opciones con los
los instrumentos de Mercado de Dinero y regresa el
instrumento seleccionado o cadena nula.
Command : Comando de salida de la función para saber si
hubo selección o no
=====
```

```
=====
Function GetNombre
Regresa el nombre del programa a ejecutar es decir
regresa : Tasas, Posicion, Result
Parametros
S Llave del Menu
Prm Parametro del programa
C2 Tipo de Operacion
=====
```

```
=====
Function NombreNuevo
Verifica si el archivo de Swapping Existe y si existe
crea otro nuevo archivo crea otro, para efecto de uso en
red.
Valor que regresa : Nombre del archivo de Swapping.
=====
```

```
=====
Procedure Ejecuta
Ejecuta un programa externo
Nombre Nombre del programa a ejecutar
Prm Parametros para el programa a ejecutar
=====
```

```
=====
Function InitMenu
  Inicializa el Objeto del Menu
  LibName  Nombre de la librería que contiene el Menu
  ObjName  Nombre del Objeto
  M        Apuntador al Menu
=====
```

```
=====
Procedure _Instrumento
  Ejecuta la funcion GetInstrumento y el Procedimiento
  ejecuta
    SM Llave de la opción seleccionada en el Menu
=====
```

```
=====
Programa Principal
=====
```

1. Verifica si existe memoria extendida para realizar el swapping a memoria extendida o a disco
2. Inicializa el archivo de Swapping
3. Inicializa el Objeto Menu
4. Procesa el Menu
5. Ejecuta la opción seleccionada
6. regresa al paso 4 en caso de que no sea el comando de salida.

CONCLUSIONES

CONCLUSIONES

Al tener la oportunidad de realizar el código del sistema utilizando la tecnología de programación orientada a objetos, nos dio la oportunidad de poder utilizar una de las herramientas de cómputo que permite crear programas de forma sencilla, controlada y de fácil mantenimiento. Con la ayuda de las estructuras llamadas objetos, que solamente se le asignan o se le eliminan atributos, se van construyendo los programas gradualmente y a la medida que el usuario final desea.

Al realizar el análisis del sistema nos encontramos con la necesidad de encontrar una herramienta que nos ayudara a elaborar las pantallas finales con la calidad gráfica necesaria para que el usuario no se confunda al momento de tomar las decisiones. Lo cual nos llevo al camino de elaborar nuestras propias herramientas gráficas; este hecho nos permitió explotar de una manera adecuada los componentes de la computadora, así como el uso de las técnicas de graficación por computadora más actuales.

La creación modular de éstas herramientas de graficación, permite utilizarlas no solamente en este sistema, sino en cualquier otro sistema que sea desarrollado con el lenguaje utilizado.

El sistema ha cumplido con el objetivo planteado al inicio del presente trabajo, que es el proporcionar una herramienta de apoyo a los operadores del mercado financiero para la toma de decisiones, ya que la funcionalidad del sistema permite observar el comportamiento del mercado financiero y con ello tener la oportunidad de obtener mayores ganancias tanto para los dueños del capital, como las personas que administran dichos bienes.

Debido a que el sistema fue desarrollado con la técnica de programación orientada a objetos, la cual, entre otras características, es que permite programar en forma estructurada, permitiendo así poder realizar relativamente fácil el mantenimiento del sistema, ya que si se requiere adaptar el sistema a nuevas necesidades, solamente se tiene que cambiar el módulo correspondiente y recompilar el sistema, sin tener que modificar todo el código.

Además, el sistema está lo suficientemente documentado para que cualquier persona ajena al sistema pueda entender el código y proceder al mantenimiento.

Como todo sistema de computación, el trabajo no termina cuando el sistema es liberado y puesto en producción. Siempre surgen nuevas modificaciones y adaptaciones para que el sistema no se vuelva obsoleto y pierda su funcionalidad, y este no va a ser la excepción, ya que el sistema puede seguir siendo mejorado. Un trabajo a futuro sería adaptar el sistema para que trabaje cien por ciento en ambiente windows, ya que esta plataforma es la tendencia para el desarrollo de sistemas en un futuro, que ya se empieza a vivir.

BIBLIOGRAFIA

Andreas Philippakis, Leonard J. Kazmier.
Information Systems Through COBOL.
McGraw-Hill Kogakusha, Ltd Tokio, Japan
1978.

Novell.Netware Getting Started: User's
Guide. Novell Incorporated. Provo,
Utah, USA, 1988.

Novell.Netware Command Line Utilities.
Novell Incorporated. Provo, Utah, USA,
1988.

Novell.Netware Getting Started:
Supervisor's Guide. Novell Incorporated.
Provo, Utah, USA, 1988.

Novell.Netware Supervisor Reference.
Novell Incorporated. Provo, Utah, USA,
1988.

Donald Hearn, M. Pauline Baker.Graficas
por Computadora. Prentice Hall, México,
1988.

TANDEM.Cobol85 Reference Manual. volumen
1 y 2. TANDEM Software Publications.

TANDEM.NonStop VLX System Management and
Operations Guide. TANDEM Software
Publications.

Borland.Turbo Pascal 5.5 Reference
Guide. Borland Publications. Scotts
Valley, CA, U.S.A., 1989.

Borland.Turbo Pascal 5.5 User's Guide.
Borland Publications. Scotts Valley, CA,
U.S.A., 1989.

Borland.Turo Debugger User's Guide.
Borland Publications. Scotts Valley,
CA., U.S.A, 1989.

Ben Ezzell. Objet-Oriented Programming
in Turbo Pascal 5.5. Adison-Wesley.
U.S.A.,1989.