



UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA



INTEGRACION DE SISTEMAS DE USUARIO EN
EXCEL Y SMALLTALK APLICANDO RUTINAS DDE.

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A :

ALICIA PINEDA RAMIREZ

Incluye un disquete 3 1/2

DIRECTOR: ING. MARIO RODRIGUEZ MANZANERA
CODIRECTOR: ING. MA. JAQUELINA LOPEZ BARRIENTOS

MEXICO, D. F.

MARZO DE 1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

DEDICATORIA:

A toda mi familia: mis padres por permitirme lograr una meta más en mi vida, otorgándome todo su apoyo y dejarme seguir mi camino de la forma correcta siguiendo los principios que me han inculcado, a Vero por ser mi hermana, a Uriel para que también logres una carrera universitaria. Los quiero.

A Manlio por compartir conmigo el periodo más hermoso de mi vida, la Universidad, por ser mi amor, compañero y gran amigo. Te amo.

A todos mis amigos que estuvieron conmigo estos 5 años.

Alicia Pineda Ramírez
México, D.F., Marzo de 1997

AGRADECIMIENTOS:

Al Ing. Mario Rodríguez Manzanera, por aceptar guiarme en la elaboración de esta tesis.

A la Ing. Jaquelina López Barrientos por su asesoría.

A Andrés Pineda y Norberto Ortigoza, por su ayuda para elaborar mi tesis y mostrarme el camino a seguir.

A todos mis profesores de la facultad, por su tiempo, y por transmitirme sus conocimientos.

INDICE

INTRODUCCIÓN	i
GENERALIDADES	ii
OBJETIVO	ii
CONTENIDO	iii
CAPÍTULO I: ANTECEDENTES DE HOJAS ELECTRÓNICAS	1
I.1 GENERALIDADES	2
I.2 SUPERCALC	3
I.3 MULTIPLAN	4
I.4 LOTUS 1 2 3	7
I.5 QUATTRO PRO	13
I.6 EXCEL	17
I.7 EXCEL VERSUS OTRAS HOJAS DE CÁLCULO	20
I.8 REFERENCIAS DEL CAPÍTULO	21
CAPÍTULO II: PLANTEAMIENTO DEL PROBLEMA	23
II.1 PLANTEAMIENTO	24
CAPÍTULO III: RUTINAS DDE	31
III.1 INTRODUCCIÓN	32
III.2 CONCEPTOS BÁSICOS	33
III.3 RUTINAS DDE EN EXCEL	39
III.4 RUTINAS DDE EN SMALLTALK	43
III.5 REFERENCIAS DEL CAPÍTULO	48

CAPÍTULO IV: DOCUMENTACIÓN DEL SISTEMA	49
IV.1 INTRODUCCIÓN	50
IV.2 DESCRIPCIÓN DEL PROGRAMA	51
IV.3 DIAGRAMA DE FLUJO	54
IV.4 LISTADO DEL PROGRAMA	56
IV.5 REFERENCIAS DEL CAPÍTULO	64
CAPÍTULO V: PRUEBAS DEL SISTEMA	65
V.1 METODOLOGÍA PARA LAS PRUEBAS	66
V.2 PRUEBAS	67
CAPÍTULO VI: RESULTADOS	71
VI.1 SOLUCIONES AL PROBLEMA PLANTEADO	72
VI.2 RESULTADOS GENERALES	75
CONCLUSIONES	77
INTRODUCCIÓN	78
APÉNDICE: MANUAL DE USUARIO	81

INTRODUCCIÓN

INTRODUCCIÓN

GENERALIDADES.

Excel es una aplicación la cual se puede explotar de la forma adecuada para obtener un reporte tanto presentable como correcto, y tiene una variedad de usos, puede ser utilizada para reportes financieros, contables, económicos etc., pero debe usarse en forma correcta, además se utiliza bajo un ambiente conversacional con el usuario (Windows), y con ello el usuario obtiene diversos beneficios, entre éstos se encuentra el uso de las rutinas DDE (Dynamic Data Exchange "Intercambio Dinámico de Datos").

Smalltalk es un lenguaje de programación orientado a objetos, el cual se encuentra bajo ambiente Windows, y da la facilidad de tener entre sus beneficios las clases que contienen las rutinas DDE. Para tener la facilidad de elaborar interfaces que permitan el intercambio de datos entre aplicaciones bajo ambiente Windows.

OBJETIVO.

Este trabajo tiene como objeto monitorear actividades realizadas sobre una o más hojas de cálculo de Excel enlazadas con Smalltalk usando las facilidades de las rutinas DDE a fin de dar seguridad al usuario en la elaboración de sus reportes.

CONTENIDO

Capítulo 1: se plantean los antecedentes de Excel, una introducción al manejo de las diversas hojas electrónicas, su inicio y evolución hasta su última versión (Excel 5.0). También se presentan las ventajas de Excel con respecto a otro tipo de hojas electrónicas.

Capítulo 2: se describen algunos de los problemas con los que se enfrenta el usuario al utilizar una hoja electrónica.

Capítulo 3: se plantea el concepto de rutinas DDE, manejadas tanto en Excel como Smalltalk, y la forma en que ambas aplicaciones utilizan estas rutinas.

Capítulo 4: en este capítulo se tiene toda la información del sistema de monitoreo, la descripción del programa que se encarga de ejecutar el sistema, la forma en que fue elaborando, así como los planteamientos y el diagrama de flujo que fueron tomados en cuenta para el inicio de la programación. Se tiene también el programa fuente que integra al sistema de monitoreo.

Capítulo 5: se tienen la pruebas iniciales y finales que se hicieron para comprobar la correcta ejecución del sistema, así como las metodologías utilizadas para la elaboración de las pruebas, y la descripción de éstas.

Capítulo 6: se presentan los resultados obtenidos de las pruebas generadas en el capítulo anterior. Las soluciones que se dan en esta tesis para los problemas a los que se enfrenta un usuario al utilizar una hoja electrónica, así como los resultados generales del sistema de monitoreo.

Conclusiones: se presentan las conclusiones generales que se desprenden de este desarrollo.

INTRODUCCION

Con este sistema de monitoreo se pretende obtener beneficios adicionales para el usuario en el manejo de hojas de cálculo desarrolladas con Excel.

Como parte final de este trabajo se presenta un apéndice que contiene el manual de usuario del sistema, el cual será una guía para el manejo del sistema de monitoreo.



**ANTECEDENTES DE HOJAS
DE CÁLCULO ELECTRÓNICAS**



ANTECEDENTES DE HOJAS DE CÁLCULO ELECTRÓNICAS

1.1 GENERALIDADES.

Las hojas de cálculo son la herramienta ideal para elaborar una serie de cálculos encadenados que den como resultado la resolución de cualquier problema, la pantalla se comporta como una ventana que se puede mover por la hoja a criterio del usuario. Las hojas de cálculo disponen de una serie de funciones auxiliares que permiten la manipulación de filas y columnas de una manera global, la realización de cálculos matemáticos complejos y la posibilidad de añadir, copiar o borrar cualquier dato que se desee.

Las hojas de cálculo tienen ciertas características en común que son:

- Tamaño de la hoja
- Sencillez de uso
- Funciones incorporadas
- Posibilidad de relaciones entre hojas
- Posibilidad de relación con otros programas.

La diferencia fundamental entre el contenido de una hoja de cálculo electrónica y el de un libro contable es que los cálculos transcritos en este último han sido previa y laboriosamente calculados, mientras que el conjunto de los cálculos de la hoja electrónica se reactualiza en cada nueva operación, por otro lado un libro contable ordinario no suele tener más de una docena de columnas y de una veintena de líneas, y una hoja electrónica tiene 4095 líneas

y 255 columnas, este número puede aumentar o disminuir dependiendo del tipo de hoja y versión que se utilice.

1.2 SUPERCALC

Debido a que muchas casas comerciales y negocios estaban adquiriendo micros cuyo sistema operativo estaba basado en CP/M, esta situación abrió las puertas al programa *Supercalc*, desarrollado por Sorcim Corporation, para convertirse en la única hoja de cálculo basada en ese sistema operativo, esto sucedió en 1980.

Supercalc es una hoja electrónica de cálculo de procesamiento visible, dispone de 64 columnas y 254 filas, al igual que VISICALC y 1-2-3. *Supercalc* utiliza el signo / (slash) para activar la línea de comandos.

Supercalc hace automáticamente cualquier cálculo que se necesite al instante o de forma repetitiva, además de que vuelve a calcular toda la hoja de trabajo cuando se cambian los valores iniciales, tiene capacidades científicas y de ingeniería, incluidas funciones matemáticas, trigonométricas y lógicas. Tiene la capacidad de proteger lugares de la hoja de trabajo que no se quieren cambiar accidentalmente, puede también preparar un informe con secuencias diferentes, poner cantidades con puntos y visualizar fechas.

En 1982 aparece la versión *Supercalc* 1.1, la cual es una versión para MS-DOS, en esta misma versión se añaden las ordenes */eXecute* y */Save part*, en ese mismo año aparece también *Supercalc* 1.12 la cual es igual a la 1.1 pero creada para el sistema operativo CP/M, después en 1983 sale a la venta *SuperCalc2*, se añade clasificación de funciones de fecha y calendario, de reanudación de ejecución y el programa SDI de conversión de datos, en ese mismo año aparece *SuperCalc3*, en la cual se ha añadido gráficos para pantalla, trazador e impresora, gestión de datos, entrada de textos sin comillas y otras funciones.

En la figura 1 se presenta la pantalla de *Supercalc* que muestra tres líneas de información del estado del programa, la línea de estado que siempre muestra la localización de la celda activa, la línea de mensaje que visualiza cuestiones y opciones cuando el *Supercalc* le indica que ingrese algo, y la línea de entrada, donde aparece la información cuando se ingresa.

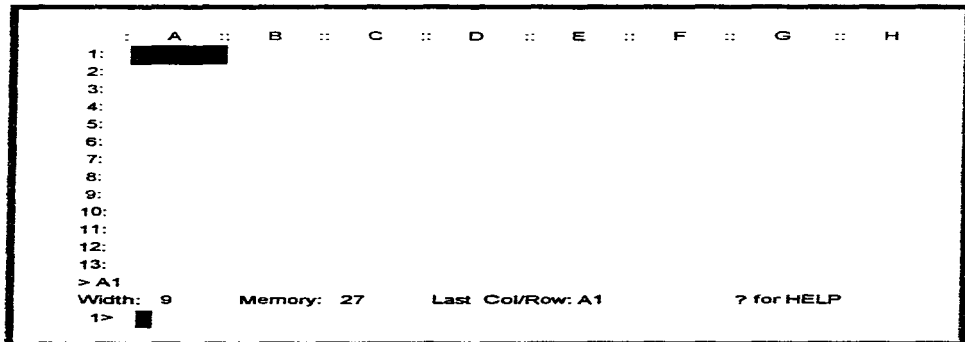


FIGURA 1: PANTALLA DE SUPERCALC.

1.3 MULTIPLAN.

La compañía que lo desarrolló se llamaba Personal Software, fue ésta la que contactó a dos estudiantes de la Harvard Business School, que, con el único fin de que les sirviese de ayuda en sus estudios, habían desarrollado un curioso programa: en éste, la pantalla aparecía dividida verticalmente en columnas ordenadas alfabéticamente; y horizontalmente en filas por orden numérico. Había nacido la Hoja de Cálculo.

Era finales de 1979 y se les ocurrió llamarlo "calculadora visible" y lo comercializaron como VISI-CALC, estaba escrito para el ordenador Apple II.

Tres años después, una importante compañía, que debía su éxito a su versión de BASIC (el MSBASIC), presenta su hoja de cálculo que llamó *Multiplan*. Los tres objetivos que se propuso cuando se abordó este proyecto fueron:

- Mantener la estructura de VISICALC.
- Mejorar cada detalle siguiendo un esquema más profesional.
- Ofrecer algo diferente.

Multiplan es una herramienta de procesamiento de hojas de cálculo en sentido de que cambia, de forma automática e instantánea, sobre los resultados individuales y generales del conjunto, dependiendo de la entrada producida de uno o varios datos nuevos en cualquier zona de trabajo, conservando las relaciones establecidas inicialmente.

Multiplan fue pionero en la idea de incluir pantallas de ayuda que guiasen al usuario hacia la correcta utilización del programa. A lo largo de los años ha tenido 3 versiones: 2.0, 2.11, y 3.2 que se han desarrollado paralelamente al MS-DOS. A partir de su versión 2.0 incluye ocho funciones puramente financieras, además de que puede emplearse desde WINDOWS.

En cuanto a la hoja de *Multiplan* presentada en la figura 2, ésta tiene 16,065 celdas distribuidas en 64 columnas y 255 líneas, su característica principal es que trabaja en memoria y no en disco.

A partir de la versión 2 amplía su tamaño hasta 4095 filas por 255 columnas. En la versión 3 se suscribe el estándar de paginación de memoria EMS. En su ambiente tiene en la parte inferior de la pantalla un menú que contiene una lista de comandos que se seleccionan con el uso de las teclas de dirección o presionando la tecla correspondiente a la primera letra del nombre

del comando. Proporciona datos sobre la memoria utilizada y disponible, así como una ventana a la cual se puede seleccionar las materias sobre las cuales se desearía informaciones complementarias.

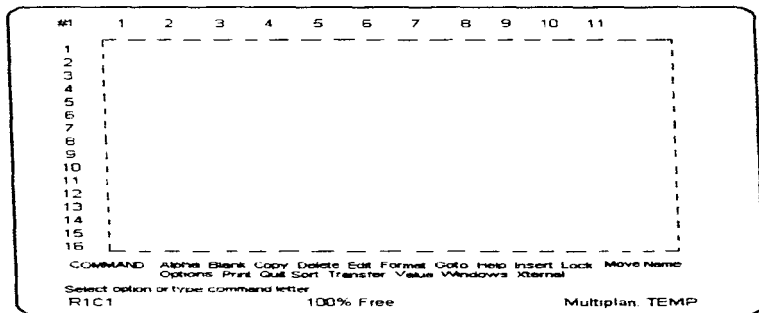


FIGURA 2. HOJA DE MULTIPLAN.

FUNCIONES DE MULTIPLAN.

- Repercute instantáneamente en un dato o un parámetro en varias hojas.
- Reestructura la disposición de una hoja añadiendo, modificando y suprimiendo una u otra zona.
- Ensancha o estrecha a voluntad el encuadre de los datos, durante o después de la creación de la hoja.
- Duplica un conjunto de valores y de expresiones aritméticas en una misma hoja o en dos diferentes.

- Organiza un modelo de simulación sin preocuparse del orden en el que se van a realizar los cálculos.
- Utiliza una parte o la totalidad de los datos para trazar gráficos.

Otras funciones que ofrece son: funciones estadísticas, matemáticas, trigonométricas, alfanuméricas y lógicas, las cuales están destinadas a efectuar una operación predeterminada.

Multiplan permite la orden de DEFINIR NOMBRE que permite identificar un grupo de celdas por un solo nombre, que servirá de argumento en una expresión aritmética cuando una de estas celdas hubiere de tomarse en cuenta para la ejecución de la operación definida.

También permite la PROTECCIÓN DEL DOCUMENTO que tiene un doble objeto: evitar la introducción o modificación de datos en algunas celdas claves y permitir el acceso a celdas destinadas exclusivamente para recibir datos ocasionales o periódicos para la reactualización de una parte o todo el documento.

1.4 LOTUS 1 2 3.

Lotus aparece en 1983, su nombre 1-2-3 viene de la combinación de tres programas de aplicación gerencial: la hoja electrónica de cálculo, la generación de gráficos, y la creación y manejo de bases de datos, la hoja de cálculo está concebida tanto para expresar llanamente un número o una frase en cada celda para llegar a construir cálculos.

Fue desarrollado por Mitch Kapor fundador de Lotus Development Corp. y Jonathan Sachs vicepresidente de desarrollo e investigaciones de dicha corporación. El diseño general de 1-2-3, la sintaxis de sus comandos y sus funciones están basadas en los convencionalismos utilizados por VISICALC.

Los elementos de su pantalla presentados en la figura 3, son los siguientes: las tres líneas superiores de la pantalla se denominan *panel de control*, este panel se utiliza para controlar la mayor parte del trabajo, cuando se realizan anotaciones en la hoja, éstas aparecen en el panel de control; además funciona como un registro de posición, permitiéndole conocer la posición actual sobre la hoja.

La línea superior del panel de control visualiza la posición del cursor en su extremo izquierdo, esta posición se conoce como dirección de celda. El extremo derecho de la línea superior del panel de control contiene un indicador de modo. La siguiente línea listará todas las opciones del menú que se haya elegido; la tercera línea muestra las opciones del menú.

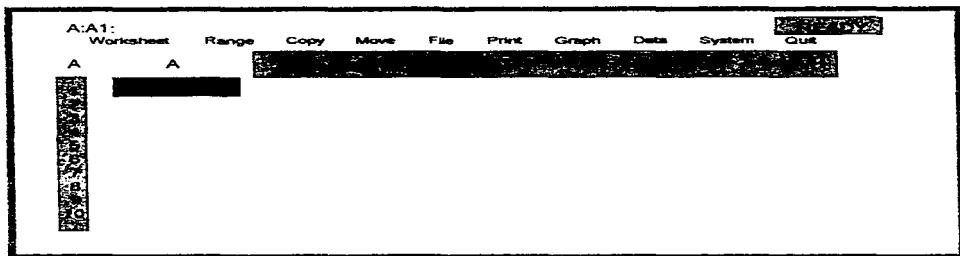


FIGURA 3: PANTALLA DE LOTUS 123.

En *Lotus* se pueden hacer los cálculos básicos que se utilizan en otras hojas de cálculo, posee además tres operadores compuestos que pueden ser utilizados con fórmulas lógicas. Estos operadores se emplean para negar una expresión o para unir dos expresiones diferentes. Los operadores son: #NOT#

CAP. I. ANTECEDENTES DE HOJAS DE CALCULO ELECTRONICAS

(#NO#), #AND# (#Y#) y #OR# (#O#). Con el uso de estos operadores se pueden añadir una segunda condición a los cálculos.

Las funciones de *Lotus* son: funciones estadísticas, la función @SUM que totaliza una lista de valores, funciones de fecha y hora, funciones de cadena (permiten trabajar con toda la entrada de rótulo de una celda o sólo con una parte de ella), funciones matemáticas, funciones financieras (se utilizan para el control de préstamos, anualidades y flujos de caja a lo largo de un período de tiempo), y funciones lógicas.

Otras aplicaciones de *Lotus* son la creación de bases de datos, la captura de los datos que la integran y las funciones que se utilizan en un programa de base de datos normal: modificaciones, ordenaciones, búsquedas, etc.

La principal aplicación del *Lotus* es la creación de MACROS, son una forma de automatización de las selecciones que se vienen realizando en los menús; en su forma más compleja, ofrecen un lenguaje de programación completo, basado en las instrucciones especiales del lenguaje de órdenes de *Lotus* 1-2-3. Una macroinstrucción es un conjunto de comandos y pulsaciones de teclas, incluidos en un programa que se almacena en la hoja de trabajo, y que se ejecutan apretando simultáneamente la tecla ALT y una letra, que es la que identifica a una macro.

El fin primordial de una macro es, al igual que el de un programa normal de computador diseñado como un lenguaje de los habituales, ahorrar trabajo al operador en las tareas repetitivas.

La versión 2.2 mostró nuevas facilidades que aumentan la utilidad del producto, por ejemplo la opción Settings Sheets (hoja de especificaciones) proporciona una panorámica de todas las selecciones que se hallan dentro de una área determinada de 1-2-3, además que a partir de esta versión, la hoja tiene 8192 renglones y 256 columnas.

La versión 2.3 incluyó la prestación WYSIWYG que permite visualizar el texto y los gráficos juntos y cambiar tipos de letra y ver el efecto. La versión 2.4 carga automáticamente el programa WYSIWYG y utiliza paneles de iconos que proporcionan una alternativa de menús.

Las versiones 2.3 y 2.4 utilizan recuadros de diálogo como alternativa para el uso de otras órdenes de 1-2-3, actualizando la información visualizada en el recuadro de diálogo.

En cuanto a los gráficos, la versión 2.2 mejora significativamente la calidad de éstos, las versiones 2.3 y 2.4 fueron diseñadas para aprovechar los monitores de alta resolución y obtener así gráficos más detallados.

La versión 3 de Lotus 1-2-3 fue diseñada para tomar ventajas de las más recientes mejoras en los chips de hardware. Adicionalmente, corre bajo las más recientes versiones de DOS y OS/2, esta versión introduce un impresionante conjunto de funciones nuevas, reteniendo la interface visual. Se agrega el uso de rangos de celdas, para la realización de una operación sobre un grupo de celdas. También permite definir varias hojas de trabajo dentro del mismo archivo, así como el uso de éstas al mismo tiempo. Esta versión fue escrita en C.

Tiene la capacidad de introducir directamente fechas y horas, convirtiéndolas en el número de serie de fecha y hora apropiado para almacenarlas.

En la versión 3.1 además de contener todas las prestaciones de las versiones anteriores, incluye un nuevo WYSIWYG, que proporciona la posibilidad de edición de la hoja. Esta versión soporta completamente la memoria expandida y extendida. Dispone de recursos propios para mejorar el sistema operativo, de forma que no sea necesario recurrir al OS/2.

Dispone de hoja en 3-D que le permite tener hasta 256 hojas en un archivo, donde cada hoja puede contener diferente información o hacer que contengan esta misma. Además de que puede traer a la memoria más de un archivo a la vez; algunas de estas hojas pueden ser archivos separados.

La versión 3.1 incluye las prestaciones de búsqueda y sustitución, las cuales son efectivas en las anotaciones de rótulos y fórmulas, y las puede utilizar para localizar una entrada en particular.

Esta versión proporciona trece funciones nuevas, más la mejora de tres funciones existentes que soportan los rangos tridimensionales. Puede sumar uno de estos rangos o hacer referencia a argumentos de función de otras hojas o archivos externos.

En cuanto a gráficos, ya no hay la necesidad de grabarlos para su impresión, ahora se imprimirá el gráfico actual o cualquier otro gráfico almacenado. También se tienen opciones que permiten girar el gráfico, cambiar el tamaño de la imagen o cambiar la intensidad de la impresión.

En cuanto a Macros, su cambio consiste en la incorporación de un registrador de pulsaciones automático, que siempre se encuentra activado, es decir que registra todo lo que se teclee sin ninguna acción especial por su parte. El registrador puede copiar las entradas en una celda y utilizarlas como una macro, también puede repetirlas para hacer tareas temporales.

La versión más actual es la de *Lotus* para *Windows* y puede funcionar tanto en un microordenador individual como en un entorno de red. *Lotus* ha sido adaptado al entorno de ventanas, menús y ayuda interactiva que *Windows* ha implantado. Una de sus características, es el uso de la paleta de iconos o *SmartIcons*, que es un conjunto de símbolos gráficos enmarcados en recuadros y reunidos en una fila o columna sobre la ventana, que permiten desencadenar las órdenes más habituales de *Lotus*.

CAP. I: ANTECEDENTES DE HOJAS DE CALCULO ELECTRONICAS

Lotus para Windows permite mantener abiertos varios documentos al mismo tiempo, cada uno de ellos encuadrado en su ventana correspondiente. Sin embargo, sólo uno de ellos se encuentra activo. Es posible definir documentos matriciales tridimensionales, al entrelazar entre si varias hoja que constituyen un archivo de datos.

Sobre la elaboración de gráficos, *Lotus* para Windows utiliza *gráficos sesgados* de la totalidad de la información y *visiones globales o fragmentarias* de la matriz de datos. También posee la facultad de leer archivos gráficos escritos en formatos PIC y CGM, y añadir así, objetos y gráficos elaborados en ambientes diferentes al *Lotus* para Windows.

Es aquí en *Lotus* para Windows donde se utilizan los enlaces interaplicaciones mediante el programa de *intercambio dinámico de datos (DDE)*, se crean enlaces permanentes entre archivos elaborados mediante *Lotus* y cualquier otro fichero incluido en aplicaciones del entorno Windows que reconozca el empleo de DDE. La creación, actualización o supresión de enlaces a través de este programa requiere que todas las aplicaciones y ficheros involucrados estén abiertos en el momento de la operación.

Otra de las facilidades de *Lotus* para Windows es la protección de la información mediante contraseñas, protegiendo desde una parte de la información de la hoja hasta la protección de un archivo completo.

En esta versión las macros son capturadas en una ventana llamada *Transcriptor*, la cual es una grabadora/reproductora de las pulsaciones que el usuario produce durante su trabajo en la sesión de *Lotus* 1-2-3.

1.5 QUATTRO PRO

A finales de 1987, Borland International introduce la primera versión de *Quattro*, la cual ofrecía significativas mejoras respecto a Lotus 1-2-3, además de sus características de recálculo en segundo plano de la información numérica, de ordenes para bases de datos y de gráficos, y de sofisticadas posibilidades de macros.

Quattro Pro combina las bases de la primera versión de *Quattro*, la tecnología de Surpass (rival de hojas de cálculo muy aclamado por sus ventanas y sus habilidades de enlace), un gran número de herramientas de dibujo y gráficos, y el programa VROOMM (Gestión de memoria virtual orientada a objetos en tiempo real). En esta versión de *Quattro* se encontrarán hojas de cálculo múltiples y ventanas, incluyendo una mayor variedad de gráficos, además de una completa utilidad para edición de gráficos. Los gráficos y los diseños pueden insertarse directamente en las hojas de cálculo.

Ahora hay más tipos de gráficos disponibles, se pueden crear diseños personalizados como el doble eje Y gráficos logarítmicos, además la mayoría de los gráficos pueden visualizarse en tres dimensiones. También es posible ajustar la anchura de las barras, posición del texto, encuadra los gráficos, etc.

Quattro Pro incluye un gran conjunto de fuentes, incluyendo los estilos Bitstream Dutch y Swiss para un amplio rango de tamaños. También se incluye el programa de utilidad SQZ! Para comprimir archivos en disco y el File Manager, un sistema completo para listar, organizar y mover archivos en disco

En base de datos se tiene la posibilidad de ordenar hasta por cinco claves a la vez, se puede añadir bases de datos externas, creadas con otros programas como Paradox, se puede establecer algún criterio de selección para extraer registros de una base de datos creada por otro programa.

Otra de sus facilidades es el uso de macros, las cuales se pueden almacenar en una hoja de cálculo aparte, llamada biblioteca de macros. *Quattro Pro* también utiliza la pantalla de Transcriptor para la creación de una macro.

El sistema de menús de *Quattro Pro* tiene una interfaz de señalar y seleccionar con menús desplegables que cubren temporalmente parte del área de trabajo. Se pueden abrir hasta 32 ventanas a la vez, lo que proporciona una gran capacidad para visualizar y comparar la información. Pueden copiarse y moverse datos entre ventanas, mostrar varias ventanas simultáneamente o adosarlas, superponerlas o ampliar una cualquiera.

La hoja de trabajo es muy grande, abarca 8192 filas y 256 columnas, ésta se presenta en la figura 4. En la parte superior se encuentran el menú principal y la línea de entrada, la línea de estado proporciona información acerca de las teclas BLOQ MAYUSC, BLOQ NUM Y BLOQ DESPL. El indicador de modo muestra mensajes sobre la tarea actual.

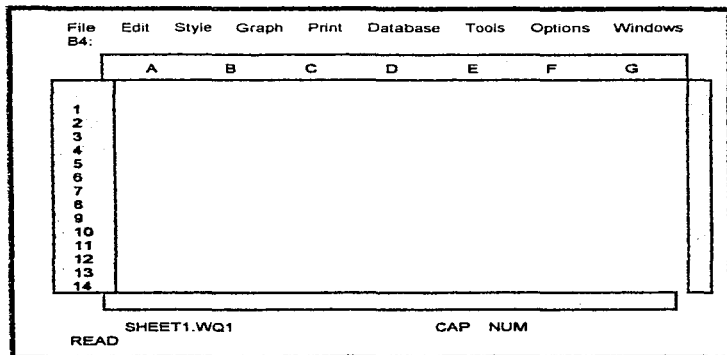


FIGURA 4: HOJA DE QUATTRO PRO.

Quattro Pro 3.0 es la última versión basada en MS-DOS, se diseñó para ejecutarse sobre un amplio abanico de diferentes sistemas y dar la mayor potencia a mayor número de usuarios.

Las características de *Quattro Pro 3.0* son similares a sus versiones anteriores, agregando la posibilidad de visualizar hojas de cálculo en la pantalla tal como aparecerán impresas, ésto mediante el programa WYSIWYG, haciendo intercambio directo de archivos con otros programas de hojas de cálculo.

En cuanto a bases de datos, se agrega la ordenación de los registros por hasta 256 claves, consultas con criterios simples o complejos. Proporciona el anotador de gráficos, un paquete de dibujo completo, con el que se resaltan

CAP. I. ANTECEDENTES DE HOJAS DE CÁLCULO ELECTRÓNICAS.

gráficos o ilustraciones; tiene un amplio rango de tipos de gráficos, y presentación de diapositivas con botones gráficos.

Quattro Pro 4.0 reproduce operaciones comunes en la hoja de trabajo, pulsando el botón apropiado en la SpeedBar, que aparece en la parte superior de la pantalla, y está compuesta de varios iconos. Ofrece nuevas herramientas para el análisis de datos gráficos, nuevas características de impresión y su manejador de impresión, nuevos comandos y características para ayudar a un mejor manejo de la sesión de trabajo como Copy Special.

Por años el sistema operativo DOS fue el único utilizado por una gran variedad de programas, pero sus limitaciones para una facilidad de uso y un mayor aprovechamiento del software aplicado, hicieron que apareciera un sistema operativo que reunía estas características, y era el Windows.

Quattro Pro para Windows toma todas las características de sus versiones anteriores, e incrementa su poder con el ambiente Windows, trae un nuevo nivel de creación y manejo de bases de datos, accedendo directamente a programas como dBase y Paradox, incluye capacidades gráficas muy sofisticadas, se pueden crear representaciones gráficas de reportes financieros y datos científicos.

Las características principales en esta versión son: la capacidad de enlazarse a otras aplicaciones de Windows, como programas gráficos o procesadores de texto, la capacidad de combinar datos con otras aplicaciones a través del Clipboard de Windows, la barras SpeedBar para desarrollar comandos comunes y procedimientos apuntando a un icono, métodos para encontrar la solución óptima a problemas complejos, y multitareas a través de Windows.

CAP. I. ANTECEDENTES DE HOJAS DE CÁLCULO ELECTRÓNICAS

Tiene una interfase gráfica (GUI), dando una variedad de letras y colores, puede utilizarse el mouse para dar dirección y movimiento a objetos, un programa WYSIWYG disponible para ver una presentación preliminar de la forma en que se imprimirán los datos.

A través de DDE y OLE ¹, puede enlazarse a otras aplicaciones de Windows. Se puede hacer una combinación de actividades mientras se está trabajando en una hoja electrónica, como desplegar la ayuda, desplegar gráficos y presentar otras hojas de trabajo simultáneamente.

La base del producto son los cuadernos con sus páginas de hojas de cálculo, cada cuaderno contiene 257 páginas, cada página del cuaderno es identificada por letras (A, B, C, ...), cada página está formada por 256 columnas y 8192 filas.

1.6 EXCEL.

Microsoft *Excel* corre bajo el ambiente gráfico de Windows y por lo tanto *Excel* absorbe todas las características del ambiente de Windows. *Excel* se divide en las siguientes áreas: hoja de cálculo, gráficos, programación, programas adicionales; soporta arreglos y tablas.

Excel tiene dos tipos de características: características de análisis, con las que se ejecutan cálculos sobre los datos de la hoja electrónica y las características de presentación, que despliega información para el usuario.

La hoja de trabajo de *Excel* es una hoja multicolumnas electrónica, tiene 256 columnas y 16384 renglones. Esta hoja se representa en la figura 5.

¹ DDE, Intercambio Dinámico de Datos, es un método de interprocesos de comunicación, consistente de intercambio de datos entre aplicaciones y sincronización de eventos. OLE Ligado e intercalación de objetos, un objeto es algo que el usuario puede manipular con una aplicación de Windows.

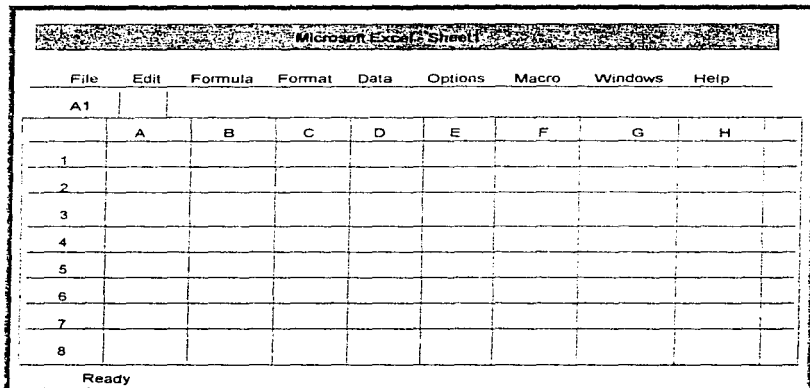


FIGURA 5. HOJA DE TRABAJO DE EXCEL.

En este caso, para la creación de una macro, se utiliza una hoja de programación de macros, y éstas se pueden ejecutar presionando dos teclas.

El enlazado de hojas electrónicas y el uso de archivos externos, la transferencia de información, exportando e importando archivos, son características principales de la primer versión de *Excel*.

Otro beneficio de *Excel* son las aplicaciones de DDE: Dynamic Data Exchange, mediante estas rutinas, se puede entablar una comunicación entre una aplicación y otra, la aplicación puede hablar con otra aplicación, mediante un lenguaje utilizando un protocolo.

CAP. I. ANTECEDENTES DE HOJAS DE CÁLCULO ELECTRÓNICAS

De la primera versión a la versión 3.0 no se sufren grandes cambios, sólo modificaciones a los comandos y el ambiente manejados por *Excel*.

Excel 4.0 está diseñado para sacar el mayor partido posible a las características de Windows y ventajas de su entorno gráfico. Los usuarios de otros programas pueden vincular sus archivos a consultas Q+E con la ayuda de un procedimiento especial de intercambio dinámico de datos (DDE). Q+E procesa las consultas y las transfiere automáticamente.

Entre las nuevas características que se encuentran en *Excel 4.0* están las nuevas herramientas y barra de herramientas para las operaciones que el usuario desee elaborar; se agregan más posibilidades automáticas (como arrastrar y soltar) para copiar y mover datos de una posición a otra, mejoras en la visualización y gestión de ventanas, los nuevos comandos de ventanas permiten efectos de un cambio que se haya producido en una sección alejada de una hoja de cálculo de gran tamaño.

Se agrega un corrector ortográfico, que comprueba la ortografía de los datos de la hoja de cálculo, así como el texto que se encuentra en los gráficos. También se tienen mejoras en la impresión, añadiendo encabezados y pies de página a la hoja, así como la integración de un asistente de gráficas, para guiar al usuario en la creación de gráficos; y para las bases de datos, un asistente para synopsis, que guía al usuario para crear una tabla desde principio a fin.

En la última versión de *Excel* (la 5), se observan cambios, primero, todas las hojas de trabajo residen en libros de trabajo y se utilizan pestañas para desplazarse a las páginas, esta característica aparece por primera vez en *Excel 4* pero sólo como una opción. La interfaz con el usuario se simplifica, con la ayuda de submenús, cajas de diálogo con pestañas y paletas que pueden desprenderse, se simplifica la clasificación y filtrado en la administración y análisis de bases de datos, así como la nueva herramienta llamada Query que permite el acceso a bases de datos externas como Paradox, FoxPro, SQL Server, etc.

Se maneja la segunda versión de OLE2 (Object Linking and Embedding) para el eslabonamiento e incrustación de objetos. Adopta el lenguaje de macros común de Microsoft basado en Visual Basic.

1.7 EXCEL VERSUS OTRAS HOJAS DE CÁLCULO

De acuerdo a la reseña anterior se puede observar que las hojas electrónicas que trabajan bajo ambiente Windows, y que por lo mismo comparten ventajas, son: Lotus 1-2-3, Quattro Pro y Excel. Lotus 1-2-3 fue una de las primeras hojas electrónicas adaptada al Windows, pero por ser una de las primeras hojas en surgir tiene sus desventajas, pues no se encuentra actualizada. En cuanto a Quattro Pro y Excel son las más actuales, aunque hay que hacer notar que la versión más actualizada y con más recursos es Excel 5.0.

Al comparar características entre una y otra (Quattro Pro y Excel), no se observan desventajas, cada una tiene sus propias prestaciones, las cuales son lo suficientemente buenas como para aceptar que ambas hojas de cálculo son excelentes para el desarrollo de la presente tesis, pero la ventaja primordial de Excel es que sus macros se pueden programar mediante el lenguaje de Visual-Basic, además del uso de las rutinas DDE.

Las rutinas de DDE, las cuales fueron implantadas desde Excel 4.0, y en Quattro Pro para Windows, se hacen notar en ambos programas, las cuales son lo suficientemente potentes para la tarea que desea realizar, y muy similares en cuanto a sus características, pero se ha escogido Excel precisamente porque es la versión más actual que se tiene, además del manejo de las rutinas DDE, las cuales son muy necesarias para cumplir el objetivo propuesto inicialmente.

REFERENCIAS DEL CAPITULO I

- Antonio Rincón
Lotus 123 para Windows
1993 pp. 15-67
Ed. Paraninfo.
- Brian Underdahl
Using Quattro Pro for Windows
Special Edition
1992 pp. 1-31
QUE.
- Burns Patrick J
Using Quattro Pro 4
1991 pp. 1-7
QUE
- C. Delannoy
Introducción al Multiplan
Barcelona 1987
pp. 2-17
Ed. Gustavo Gili
- Chris Wood
Programa SUPERCALC a su alcance
1986 pp. x-xvi
McGraw Hill
- Christopher Van Buren
Excel 4.0
1992 pp. 1-14
Anaya Ed.
- Eddie Adamis
Aplicaciones de Multiplan y ...
1995, pp 15-27
McGraw Hill

- J. López de Aguilar
Multiplan 2a. Ed. 1989
pp. 11-27
Paraninfo.
- LeBlond y Cobb
Como usar Lotus 123
1987 MFE1-MFE15, C1.12-C1.22
Macrobit
- Martin S. Matthews
Excel made easy
1990 pp. 1-15
McGraw Hill
- Mary Campbell
1-2-3 a su alcance versiones 2.2, 2.3, 2.4
1993 pp. 1-30
McGraw Hill
- Mary Campbell
1-2-3 Versión 3.1 Manual de referencia
1992 pp. 3-29
McGraw Hill.
- Stephen Cobb
Aplique QUATTRO Pro
1992 pp. XXV-20
McGraw Hill
- Stephen Cobb
Quattro Pro 3.0
1992 pp. 1-35
McGraw Hill
- Thomas Chester
Excel 5 para Windows
1995 pp. xxxiii-xxxvii
Ventura

II

PLANTEAMIENTO DEL PROBLEMA



PLANTEAMIENTO DEL PROBLEMA

Los problemas a los que se enfrenta un usuario cuando utiliza una hoja de cálculo electrónica son diversos, y claro está, su complejidad dependerá del tipo de hoja y la versión que este esté utilizando. En este trabajo se tratará de generalizar todos los problemas a los que se enfrenta un usuario, cuando se encuentre utilizando cualquier tipo de hoja electrónica, no importando su versión.

El principal problema al trabajar con cualquier hoja de cálculo electrónica es su tamaño, ya que normalmente son hojas compuestas por un gran número de celdas, denominando celda a la intersección que existe entre un renglón y una columna, por otra parte el tamaño del monitor, el cual normalmente resulta ser pequeño aún cuando se esté procesando en una estación de trabajo, ya que no es posible en la mayoría de los casos abarcar el total de la hoja, provocándose que ésta no pueda ser visualizada más que por áreas de tamaño específico seleccionadas ya sea por el usuario o por default, y que dependerán directamente del tamaño del monitor ya que al desplegarse la hoja en éste, su tamaño en columnas y renglones permanece constante sólo pudiendo variarse si se modifica el valor del parámetro ZOOM (acercamiento). Cabe señalar aquí que aún variando los parámetros únicamente se podrá ver una sección o parte de la hoja de trabajo.

A fin de dar un ejemplo, en el caso de Excel 5 que se muestra en la figura 6, el proveedor para reducir dicho problema al usuario, ha dividido su hoja electrónica en libros de trabajo, en donde cada libro está compuesto de 16 hojas de trabajo, pudiendo ser visualizada cada una por separado, esto mediante las pestañas que se encuentran debajo de cada hoja de trabajo que especifica el nombre o la hoja que se está visualizando en ese momento.

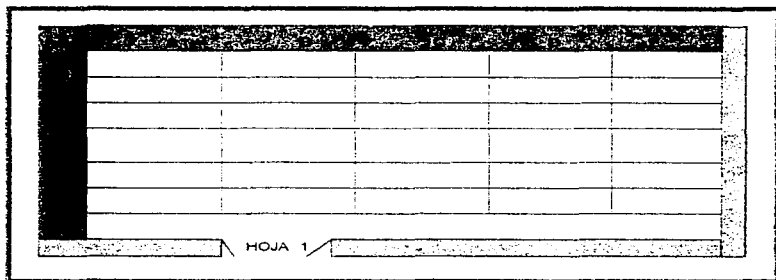


FIGURA 6. HOJA DE TRABAJO DE EXCEL 5.

También pueden ligarse las hojas, unas con otras, con la hoja en la cual se está trabajando, igualmente se pueden ligar los libros de trabajo y hacerse las mismas operaciones que se realizan cuando se trata de una hoja de trabajo única. Aún así el problema de no poder visualizarse la(s) hoja(s) en su totalidad permanece.

Debido a lo anterior en caso de que el usuario quiera observar todos los cambios ocurridos en una hoja de trabajo electrónica, ésto le será imposible, pues no puede saber con seguridad en que momento ocurre una modificación en alguna celda y en que determinada sección está ocurriendo. Ésto es válido cuando existe un proceso ejecutándose paralelamente al trabajo del usuario o bien cuando alguna de las opciones de recálculo haya quedado encendida. Suponga por un momento que se encuentra en la celda F90, y que al modificarla, se modifica también la celda A2, debido a que usted supone que ambas celdas están relacionadas de alguna manera, por supuesto usted seguirá realizando su captura confiando en la modificación ocurrida automáticamente en A2, usted creerá que su informe se encuentra sin errores, debido a que confió en que todo resultó bien, sin embargo al revisarlo se da cuenta del error cometido, todo ello derivado de que no pudo observar la hoja completa.

Las modificaciones en las celdas dependerán del formato de ellas, o en la forma en que éstas fueron editadas, uno de los cambios que se da en forma automática en una celda es debido al uso de fórmulas. Aquí, el lugar en donde se encuentra la fórmula (la celda) cambia automáticamente al darle diferentes valores a las celdas involucradas en ella, sin que el usuario pueda detener el proceso de recálculo de éstas con los nuevos valores, a menos de que vuelva a modificar el valor de la celda que pertenece al cálculo, lo que ocasiona un nuevo cambio en las celdas involucradas. Un ejemplo de ello se da al introducir la fórmula en la celda D3, que contiene $ATAN(y/x)$, "y" se encuentra en la celda C3 y "x" en B3, el valor lógicamente dependerá de los valores de las celdas C3 y B3, si el usuario cambia tan sólo una de ellas, el valor de la celda D3 estará cambiando.

Otro de los procesos de modificación que afecta directamente a las celdas que contienen una fórmula, es ir directamente a la celda involucrada e introducir un valor o bien copiar el valor de otra celda para modificarla. Supóngase que se toma un rango de celdas que no se quieren cambiar de lugar, al seleccionarlas se va al lugar que se desean copiar, pero

accidentalmente se suelta el botón de arrastre, y se deja desplazado en otro lugar, en ese momento las celdas que se encontraban allí, se han perdido y se encuentra el rango de celdas que se estaba arrastrando con la consecuente pérdida de las fórmulas de las celdas marcadas. Lo anterior se puede hacer accidentalmente, o conscientemente, de modo que la hoja quede mostrando resultados que no necesariamente correspondan a las fórmulas que deberían ocupar cada una de las celdas que conforman el reporte.

Otro problema al que se enfrenta el usuario, es el que se le presenta cuando manda imprimir la hoja que en ese momento muestra el monitor la cual ya ha sido almacenada, ésta se imprimirá como un reporte, y si se tienen fórmulas en ella, éstas no se imprimen en la hoja, ya que normalmente en el monitor se observa solamente el resultado de aplicar la fórmula, a menos de que se quiera editar ésta. Cabe señalar aquí que al posicionarse en la celda que contiene la fórmula, ésta solamente se podrá observar en la barra de mensajes, de otra forma y de no existir ésta, solo se verá el resultado. Ésto se podría prestar para simplemente capturar el informe sin necesidad de agregar fórmulas, ajustándolo a la conveniencia de la persona, y simplemente mandándolo a imprimir. Este problema es el más común de los que se presentan en la práctica, cuando una secretaria que no sabe utilizar una hoja electrónica, simplemente se dedica a introducir los datos en una tabla, de modo que queden similares al reporte que se le ha perdido, sin hacer caso de si existen fórmulas o no.

Así, dado que la persona que toma las decisiones o revisa la información del cuadro ya no revisa si las columnas y renglones cuadran, simplemente confiará en que el reporte está bien hecho, pues a su parecer se elaboró en una hoja de cálculo electrónica, debiendo estar correcta ya que así lo muestra su impresión. De ahí se derivan muchos de los problemas contables de las empresas.

Por otra parte y adicionalmente a lo anterior, al mandar imprimir la hoja, nuevamente no podrá observarla completamente, ya que solo se puede imprimir por secciones y al revisar el documento impreso, el usuario podrá igualmente confundirse, pues al observarlo no habrá gran cambio entre lo que muestra el monitor y el documento impreso, ya que se vuelve a tropezar con el mismo problema de que la hoja es demasiado grande para visualizarla o llevar a cabo los cálculos mentalmente a fin de comprobar si éstos cuadran.

Generalmente las fórmulas tampoco se imprimen, sólo se imprime el resultado de ellas, y no se puede comprobar que los cálculos fueron realmente hechos, o simplemente se capturaron los resultados finales, lo cual es válido tanto para una sola celda como para su totalidad.

Un problema que también es muy común es el derivado del ligado de hojas de trabajo, pues ahora se tiene una hoja más grande aún, con información que pertenece a un mismo reporte o problema, pero que contiene datos que son diferentes. Por ejemplo si se quiere observar el cálculo en las celdas, al ir navegando entre las hojas, el cálculo en una celda se perderá o simplemente al ir pasando de hoja en hoja, no se vuelve a la celda con la que se estaba trabajando. Si son muchas las hojas ligadas hasta el mismo usuario se confundirá, ya que si éste se descuida por un momento, al volver al trabajo, no sabrá que ha pasado, pero continuará con lo que estaba haciendo. Un caso muy común en los modelos de negocios hechos en hojas de cálculo.

También es muy posible que se quiera hacer un resumen de todas las hojas ligadas y se empiece a copiar rangos de celdas de las diferentes hojas que se encuentran ligadas en ese momento, y copiarlas todas en una nueva hoja, como se observa en la figura 7.

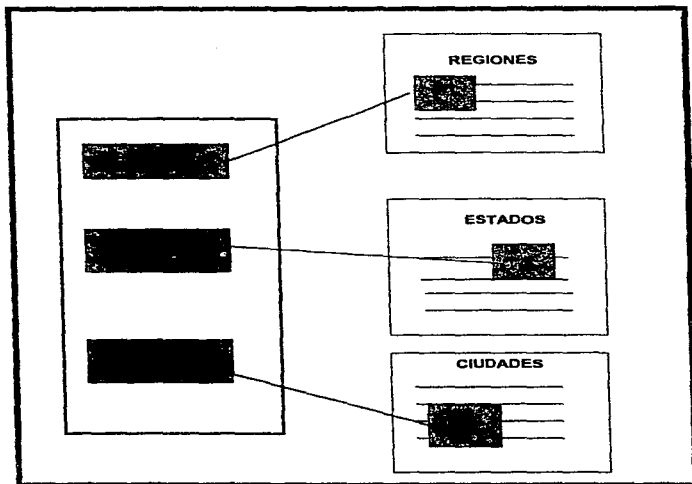


FIGURA 7: EJEMPLO DE HOJAS LIGADAS.

Si en el rango de celdas que se está copiando hay otras celdas involucradas y el usuario no las copia al encontrarse en la nueva hoja que se llamará REPORTE, entonces el rango de celdas que se copió, tomará nuevos valores; es decir si el grupo de celdas de CIUDADES necesita los valores de las celdas de ESTADOS, y en REPORTE se copian por separado, se tendrán valores diferentes, así se tendrán graves problemas al imprimir los reportes de

CIUDADES y ESTADOS por separado. El error al cual nos estamos refiriendo se descubrirá al imprimir el documento REPORTE, ya que al compararlos no cuadrarán los resultados, esto en caso de que se llegaran a revisar los documentos impresos, y no simplemente que se hagan y se pasen a las personas que los solicitan, situación común en la práctica.

Otro problema que se puede observar se presenta en el manejo de macros, pues las instrucciones que conforman una macro se graban en una hoja como si fuese un archivo más, la única diferencia es que en la pestaña de la hoja de trabajo, se podrá observar el mensaje de MODULO1, pero si se trata de un usuario que solamente se dedica a pasar reportes, y no se ha dedicado a aprender lo que implica el EXCEL, entonces pensando que se trata de una simple hoja de trabajo, se puede confundir y proponerse a hacerle cambios o modificaciones lo cual en principio está mal.

Un último problema es el ocultamiento de hojas o de un rango de celdas, esto se hace para impedir los cambios por parte de usuarios externos en cualquier hoja de cálculo ya dedicada a la producción. Si se termina Excel con un libro de trabajo u hoja oculta, ésta se ocultará automáticamente la siguiente vez que se abra. Recordar esto es sumamente importante como protección contra usuarios externos, pero en dado caso de que se olvide de que alguna hoja se ha ocultado, posteriormente no se sabrá que ha pasado con esa hoja, o porqué se obtienen determinados resultados si no existen tales celdas o fórmulas que produzcan cambios. Es sumamente importante la protección de datos, pero si se llegan a olvidar, esto ocasionará grandes dolores de cabeza. Este problema se presenta comúnmente en modelos de negocio desarrollados por casas extranjeras en México, en donde el diseñador oculta parámetros, tablas, comentarios, etc., a su cliente. Una vez liberado el sistema, el usuario no entiende o no alcanza a visualizar porqué se modifican ciertas celdas en forma automática.

III

RUTINAS DDE

III

RUTINAS DDE

III.1 INTRODUCCIÓN.

En este capítulo se presenta lo que implica para el usuario y el programador el utilizar las rutinas DDE, a fin de cubrir el objetivo primario de esta tesis. En primer lugar se tratarán los conceptos básicos de DDE, a partir de estudiar la conformación de las propias rutinas, es decir la forma en que se integran dentro de un lenguaje y sus características principales, se estudiará a través del uso de algunos ejemplos, como se puede establecer la comunicación de una aplicación con otras, así, como los tipos de comunicaciones que existen.

A fin de clarificar al lector lo más posible la comprensión de las rutinas DDE en Excel, se tratará en primer lugar la construcción de una aplicación en esta hoja de cálculo, debe tomarse en cuenta que en este primer ejemplo, la aplicación de Excel funcionará como el servidor del sistema, lo que implica que Excel puede ser utilizado tanto como servidor como cliente de una aplicación. Más adelante se definirá la forma en que funciona Excel como servidor de otras aplicaciones, y como debe y puede iniciar una conversación con ellas.

Finalmente se incluirán varias muestras de como las rutinas DDE se presentan en Smalltalk, el cual para nuestro caso siempre funcionará como el cliente del sistema, pues se dedicará a pedir los datos a la hoja de Excel necesarios para su procesamiento. Además se definirán los mensajes que debe de enviar Smalltalk utilizando sus clases hacia Excel, para la petición de

datos.

Se debe tener siempre presente que una conversación DDE toma lugar entre dos ventanas, una para cada una de las aplicaciones participantes. La primer ventana podría ser la ventana principal de la aplicación, que se encuentra asociada con un documento específico.

Una conversación DDE siempre utiliza un par de manejadores de ventanas que se encuentran ligadas en la conversación. Si una aplicación abre una ventana para establecer una comunicación y no encuentra abierta la ventana de la aplicación con la que desea comunicarse no existirá tal comunicación hasta que dicha aplicación abra su propia ventana.

Cada aplicación al intentar llevar a cabo una conversación mediante DDE debe asegurarse que solamente exista una ventana para cliente y una para servidor, es decir, no permitir que la conversación envuelva más de dos ventanas en la conversación entre las aplicaciones, una específicamente para el cliente y otra para el servidor, ésto lo logra haciendo que se congelé cada ventana durante la conversación.

III.2 CONCEPTOS BÁSICOS.

El "Intercambio Dinámico de Datos" (DDE Dynamic Data Exchange) es un protocolo con el cual Microsoft Windows permite que dos aplicaciones "conversen" para intercambiar datos continuamente y, si se desea, automáticamente.

El "Intercambio Dinámico de Datos" (DDE) es un método de comunicación entre procesos. Una comunicación entre procesos consiste en pasar datos entre ellos sincronizándolos a través de eventos. DDE usa parte de la memoria del sistema en donde se está ejecutando, para intercambiar datos entre aplicaciones, además utiliza la secuencia de pasos que constituye su

protocolo para sincronizar el intercambio de datos. DDE automatiza el proceso de cortado, pegado y copiado que un usuario lleva a cabo en forma manual para pasar información entre aplicaciones, permitiéndole una forma más rápida y automática de actualizar su información.

Las rutinas DDE son un conjunto de reglas a manera de protocolo, que todas las aplicaciones de DDE deben respetar. Si un usuario accesa a una aplicación de DDE que no respeta el protocolo DDE en forma correcta, la aplicación podría no ejecutarse bien o incluso podría suspender al sistema del usuario en su totalidad.

El protocolo DDE funciona en dos modalidades o tipos de aplicaciones a saber: el primer modo se basa en el manejo de un mensaje, mientras que el segundo es una aplicación de las rutinas contenidas en la Biblioteca de la Administración de Intercambio Dinámico de Datos (Dynamic Data Exchange Management Library, DDEML)².

A su vez las aplicaciones DDE caen en cuatro categorías diferentes: cliente, servidor, cliente/servidor, y monitor.

CLIENTE O SERVIDOR:

Una conversación DDE ocurre entre una aplicación cliente y una aplicación servidor. Una aplicación cliente solicita datos o servicios de una aplicación servidor.

CLIENTE/SERVIDOR:

Una aplicación cliente/servidor funciona como ambos, es decir como una aplicación cliente o como una aplicación servidor, de esta manera solicita o provee información.

² La aplicación DDML usa una Biblioteca de Encadenamiento Dinámico (Dynamic Link Library, DLL).

MONITOR:

Una aplicación monitor puede interceptar mensajes DDE de otras aplicaciones sin embargo la aplicación DDE no puede actuar sobre ellas.

MENSAJE DDE:

El corazón del protocolo DDE es el mensaje DDE. El protocolo define el cuándo, el cómo y en dónde el mensaje DDE es utilizado.

Los componentes de una conversación DDE se muestran en la figura 8.

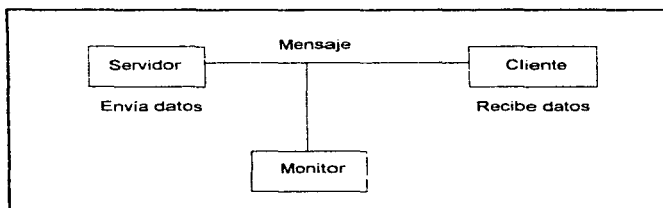


Figura 8 Componentes de conversaciones DDE

El protocolo DDE especifica que los mensajes en una conversación tienen que ser manejados de manera síncrona. Sin embargo una aplicación puede conmutarse entre conversaciones también en forma síncrona. La información generalmente fluye del servidor al cliente, aunque éste puede, si es necesario, enviar información de retorno al servidor.

Una aplicación, al igual que un televisor conectado a cablevisión, utiliza un canal para recibir (y transmitir) información, a diferencia de un televisor no conectado a este tipo de servicio, el canal DDE admite conversaciones en las que los datos fluyen en ambas direcciones. Cuando un servidor recibe una petición de conversación sobre un tema que reconoce, responde y abre un

canal. Una vez establecida la conversación, ésta no puede cambiar de tema ni de aplicación. Para comenzar una conversación con otro servidor diferente o con el mismo servidor pero sobre otro tema, se deberá iniciar una conversación nueva en otro canal. Esto no afecta a la conversación que tiene lugar en el primer canal; la cual puede finalizarse o reanudarse en forma independiente.

En Microsoft Access un manejador de bases de datos de Microsoft, se puede iniciar una conversación con una aplicación por un canal y responder a otra aplicación por otro canal. En esta situación, Microsoft Access es el cliente de la primera conversación y el servidor de la segunda.

NOTA: Hasta el momento Microsoft Access es la única aplicación que puede funcionar como cliente y servidor al mismo tiempo.

Las aplicaciones DDE usan tres sistemas de identificación para distinguirse entre otras aplicaciones:

1. El nombre (*name*) de la aplicación que define una aplicación servidor estándar.
2. El nombre del tópico (*topic name*) el cual define una aplicación servidor.

Las aplicaciones servidor pueden soportar uno o más tópicos o temas (*topics*). Los temas definen el asunto de la conversación y son como una unidad de datos significativa para la aplicación servidor. Para la mayoría de las aplicaciones que actúan sobre archivos, se trata de un nombre de archivo. Cada tema puede tener uno o más nombres (*item names*), los cuales identifican los detalles de un *topic name*. Un *item* es una referencia a un fragmento de datos (como un rango de celdas de una hoja de cálculo de Excel) que se puede intercambiar entre dos aplicaciones. El *item* hace referencia a un fragmento determinado de un tema.

3. El "procesamiento de mensajes" en el orden correcto. Lo cual es de suma importancia para que las aplicaciones envíen mensajes en un tiempo apropiado.

Una conversación DDE comienza cuando una aplicación cliente inicia una conversación enviando un mensaje a una aplicación servidor. Dicho mensaje debe contener el nombre de un tema (*topic name*). También para comenzar una conversación deberá existir una aplicación servidor que responda al mensaje. Así cuando la aplicación cliente envía el mensaje, éste pasa a su manejador de ventanas y especifica el nombre de la aplicación servidor y el tópicó para la conversación.

Después de establecida una conversación DDE, las aplicaciones cliente y servidor comienzan su trabajo real: de intercambiar datos y desarrollar servicios. Intercambiar datos solamente parecería apropiado para una aplicación de intercambio dinámico de datos, sin embargo la aplicación servidor también puede desarrollar este servicio. Por ejemplo: un servidor estando conectado a una computadora podría estar abriendo un archivo y bajándolo a ésta al mismo tiempo.

Cada mensaje es pasado hacia la aplicación servidor como una cadena de caracteres que contiene un comando para ésta. El protocolo DDE siempre define el formato para dicha cadena, no así su contenido.

El intercambio de datos puede ocurrir en tres formas distintas:

1. Una aplicación cliente puede solicitar datos a una aplicación servidor en un tiempo base.

En este caso, la aplicación cliente envía un mensaje, luego la aplicación servidor agradece el pedido retransmitiéndole el mensaje. Si la aplicación servidor responde con un mensaje, la aplicación cliente puede acceder el dato. Por el contrario si la aplicación servidor no puede responder a la petición, éste envía una respuesta negativa junto con el mensaje.

2. El cliente envía datos a un servidor.

Cuando ésto ocurre, el cliente envía los datos al servidor usando un mensaje. Cuando el servidor recibe este mensaje, responde al cliente con otro mensaje especificando que acepta el dato.

3. El servidor advierte al cliente que un ítem ha cambiado de valor.

Esta petición puede tomar dos formas. En la primera, el servidor envía una notificación de que el dato ha cambiado, pero no envía el dato. En la segunda, el servidor envía el dato cada vez que éste cambia. Es decir el cliente solicita enviar un mensaje al servidor, el servidor responde a este mensaje con otro mensaje. Entonces cada vez que el dato cambie, el servidor enviará un mensaje conteniendo el dato. Si el cliente quiere solo notificación, el mensaje no incluirá el dato.

Los métodos para intercambio de datos son frecuentemente asociados con el término *encadenamiento (link)*. Dicho término denota la forma en que los datos son intercambiados. Si un cliente solicita datos y el servidor inmediatamente los envía al cliente se dice que se trata de un encadenamiento en frío (*cold link*). Un encadenamiento tibio (*warm link*) se da cuando un servidor advierte a un cliente que un dato ha cambiado de valor, pero no puede enviar el valor a el cliente, hasta que éste procese el método de solicitud (*Request*). Finalmente un encadenamiento en caliente (*hot link*) ocurre cuando el servidor envía el nuevo valor de un dato al cliente cada vez que el valor ha cambiado. Los encadenamientos en caliente transcurren en tiempo real.

Para dar por terminada o concluida una conversación entre cliente y servidor es indistinto quien dé inicio a dicho proceso. Una aplicación ya sea cliente o servidor enviará un mensaje cuando quiera dar por terminada una conversación, cuando el compañero en la conversación reciba dicho mensaje, también podrá a su vez enviar un postmensaje. Esto puede ocurrir en cualquier tiempo durante la conversación.³

³ Lo anterior debe ocurrir antes de que la aplicación solicitante llame a la función `PostQuitMessage()` y deje su mensaje principal en un ciclo (loop).

III.3 RUTINAS DDE EN EXCEL.

La hoja de cálculo Excel tiene varias facilidades para importar y exportar datos, como lo son OLE (Ligado e intercalación de Objetos) y DDE (Intercambio Dinámico de Datos). Mientras las rutinas de DDE se pueden utilizar para copiar texto hacia Excel, y para copiar datos fuera de Excel. OLE, se utiliza únicamente para copiar gráficos hacia Excel.

Con DDE y OLE también se pueden hacer las mismas actividades de intercambio de datos y gráficos desde Excel hacia cualquier otra aplicación que soporte DDE y OLE.

Excel dentro de sus rutinas de DDE funciona por default (normalmente) como servidor, por ejemplo si se tiene una conversación entre Word y Excel, Word puede solicitar de una hoja de trabajo de Excel en particular, el contenido de una celda o un rango de celdas. Word también puede enviar información a ella, es decir tiene la capacidad de enviar un texto a una única celda o rango de celdas y comandos a una aplicación. Hay que recordar que para iniciar una conversación entre aplicaciones, ambas deben de encontrarse activas.

Si se utiliza a Excel como servidor, se deben tomar en cuenta los siguientes puntos:

- El nombre de la aplicación para DDE en el paquete Microsoft Excel es "Excel".
- El paquete Microsoft Excel respeta el estándar del conjunto de tópicos, el cual soporta los siguientes ítems:
 - SysItems (ítems del sistema),
 - Topics (temas),
 - Status (estados),
 - Formats (formatos),
 - Selection (selección).

Protocols (protocolos).
EditEnvitems (edición de ítems).

- Cualquier documento abierto es válido como un tópic de DDE en Excel.
- Una celda o rango de celdas es un ítem en una hoja de trabajo o en una hoja de macros.

NOTA: Al especificar a la celda o rango de celdas como un ítem, se puede usar un nombre definido de Excel que las identifique aunque también existe la posibilidad de que las indique manualmente el mismo usuario.

- Usando la instrucción DDEexecute, se pueden enviar los comandos de una macro encerrados entre paréntesis cuadrados.*
- Ejecutando una macro de Excel se pueden enviar comandos, lo cual se puede ejemplificar con las siguientes instrucciones:

Esta es una macro que le da formato a un conjunto de celdas:

```
DDEexecute chan. "[run (" + Chr$(34) + "MACROS.XLM!FormatCells" + /  
Chr$(34) + ")"]"
```

NOTA: Chr\$(34) es igual a el carácter comillas.

- No se puede inicializar un ligado de una aplicación DDE con una hoja de trabajo de Excel si la barra de fórmulas en Excel se encuentra activa. Si un usuario intenta iniciar el ligado mientras la barra se encuentra activa, se desplegará en su aplicación un mensaje de error diciendo que la aplicación no responde.

* Excel no acepta comandos de Visual Basic a través del canal de DDE.

- Excel incluye una opción que previene a otra aplicación si es que Excel está iniciando una conversación con ella.

Ejemplo de lo anterior se muestra en la siguiente macro la cual inicia una conversación DDE con MIHOJA.XLS (una hoja de trabajo de Excel).

```
Sub MAIN
chan = DDEInitiate ("Excel" "MIHOJA.XLS")
figures$ = DDERequest$ (chan "L19C1: L22C7")
Insert figures$
DDETerminate chan
End Sub
```

Si se utiliza un nombre para el grupo de celdas, por ejemplo "Estados", se puede cambiar la instrucción por:

```
figure$ = DDERequest$ (chan "Estados")
```

NOTA: Un error frecuentemente repetido en la práctica, aún por usuarios avisados en el uso de Excel, es el que se da cuando se tienen diversas versiones de Excel o de paquetería de Microsoft y se desconoce si las versiones son para el idioma español o el inglés, dado que la especificación de renglones y columnas (LC, línea y columna) no coincide con la especificación en inglés (RC, row and column).

Muchas macros en Excel regresan el valor lógico TRUE (verdadero) si están corriendo satisfactoriamente. Algunos comandos de macros en Excel, incluyendo CREATE.OBJ(), GET.OBJECT(), GET.NAME(), GET.NOTE(), y otros más iniciado con GET.<comando>, pueden regresar otros valores diferentes a TRUE (verdadero). Un macro no debe incluir estos comandos ya que no pueden ser enviados directamente a Excel a través de DDE. Estos comandos no generarán mensajes de error, simplemente no trabajarán.

Cuando se usan los comandos de ligado Copy y Paste para copiar y pegar datos de otra aplicación, el ítem o tópico en el ligado creado podrá ser hasta de 128 caracteres. La fórmula podrá ser truncada por Excel, pero los datos se desplegarán y actualizarán correctamente. Si se intenta modificar y volver a introducir la fórmula de ligado cuando el ítem o parte del tópico son más grandes de 128 caracteres, Excel truncará el ligado y desplegará un mensaje de error.

Finalmente se da un ejemplo para la inicialización del intercambio dinámico de datos (DDE) entre una macro de Excel y una aplicación de Visual Basic que se encuentre corriendo al mismo tiempo.

Cuando se inicia el trabajo con Excel, siempre se crea una hoja de trabajo por default que recibe el nombre de HOJA1.

En alguna celda, por ejemplo C2, se teclea el texto que se quiera enviar a Visual Basic y se salva esa hoja.

Se crea una nueva macro en Excel, siguiendo el formato:

```
Record1 (a)
chan= INITIATE ("dde", "hoja1")
= POKE (chan, "A:\hoja1.XLS" & C2, "texto")
= TERMINATE (chan)
= RETURN (chan)
```

La función de INITIATE inicia con éxito una conversación con la aplicación especificada, asignándola a un canal, si la aplicación especificada no está en ejecución, o si lo está pero no reconoce el tema indicado, se producirá un error interceptable. Se debe agregar código para verificar si una aplicación se está ejecutando y para iniciarla en caso de que no lo esté. Con INITIATE se arranca la aplicación de Visual Basic si éste no está corriendo y establece el ligado con la HOJA1. La función POKE pone el texto en la celda, (proporciona los datos al servidor) C2 de la HOJA1. La función TERMINATE

termina el ligado, y regresa a la función de fin de la macro.

Los canales DDE nunca se cierran automáticamente, si se utiliza una variable local para almacenar el número de canal y no lo cierra cuando termina el procedimiento, dejará el canal abierto sin posibilidad de cerrarlo. Por lo que se utiliza la instrucción DDETerminateAll que termina todas las conversaciones DDE activas.

III.4 RUTINAS DDE EN SMALLTALK.

Excepto por un pequeño núcleo (kernel) en lenguaje ensamblador, Smalltalk está escrito en Smalltalk. En este núcleo se encuentran las primitivas hechas en ensamblador, en éstas se encuentran las operaciones simples y de bajo nivel, comunicaciones con el sistema operativo o hardware. Ejemplos de primitivas son las operaciones aritméticas, booleanas, lógicas, accesos a sistema de archivos, llamadas a disco duro y floppy.

Su código fuente está organizado en módulos, los cuales le permiten al usuario poder reutilizarlo y modificarlo para su aplicación. El sistema le ayuda al usuario a reutilizar tanto código preescrito como es posible, editarlo y modificarlo rápidamente para su propio programa. Usted puede crear un programa pieza por pieza y ver los resultados inmediatamente.

En resumen, Smalltalk toma ventaja completa de Windows, específicamente de características como Dynamic Link Libraries (DLL) y Dynamic Data Exchange (DDE).

Las rutinas DDE en Smalltalk son un servicio externo de comunicación. Una aplicación escrita en Smalltalk inicia la conversación convirtiéndose desde ese momento en la aplicación cliente, mientras que la aplicación que le responde se convertirá en la aplicación servidor. El cliente hace una petición y el servidor la atiende y opcionalmente regresa la respuesta.

Dado que una aplicación servidor puede recibir peticiones de múltiples clientes, y a su vez cada cliente puede conversar con múltiples servidores sobre múltiples tópicos, para identificar una conversación única, DDE usa tres nombres identificadores, "aplicación", "tópico", e "item". Un servidor tiene solo un nombre de aplicación, al menos un topic name, y uno o más items por tópico.

Cuando una aplicación cliente hace una petición a un servidor, éste utiliza los tres nombres para identificar la conversación en la cual desea tomar parte. Durante una conversación se hace uso de una serie de mensajes, comenzando con un mensaje

DDE_ INITIATE que incluye el nombre de la aplicación y su tema (topic name).

Siempre y cuando el cliente reciba la respuesta de que el servidor soporta la aplicación y el tema, el cliente enviará una petición de mensaje con el ítem name. Si el servidor entiende el ítem solicitado, le responderá con el mismo dato; si no, responderá con un mensaje de agradecimiento negativo. Así la conversación continuará hasta que el cliente la termine.⁵

Como se explicó, hay tres tipos de conversaciones DDE: *cold link*, *hot link*, y *warm link*. El proceso anteriormente descrito es un *cold link*. El cliente hace una petición, el servidor responde con el ítem pedido. En una conversación *hot link*, si el servidor no puede dar el ítem entonces la conversación es congelada hasta que el ítem se encuentre disponible a diferencia del *warm link* el cual además sigue la opción de que el cliente comience notificando que el ítem pedido está disponible, y en su tiempo libre lo colecciona.

Para aclarar los conceptos anteriores se procederá a dar un ejemplo descriptivo que los incluya. Suponga por un momento que se lleva a cabo una conversación sobre la participación o no de una persona a una fiesta.

⁵ Por convención, cada conversación es manejada por una ventana separada que está usualmente congelada.

Suponga como primer caso que se tiene una conversación cold link. En primer lugar se requiere de una fiesta. Si no hay fiesta, el invitado buscará otra fiesta. Ahora bien en una conversación hot link, el invitado dejará una petición de ser llamado tan pronto como haya fiesta. Finalmente en una conversación warm link, el invitado notifica al anfitrión su deseo de asistir a una fiesta cuando está exista y para tal fin le deja su número telefónico.

Smalltalk soporta DDE como una clase sencilla conformada por dos subclases.

"DynamicDataExchange" es el nombre bajo el cual Smalltalk reconoce a la clase de las rutinas de DDE, la cual provee de un armazón y de un código de bajo nivel para las rutinas DDE.

Los métodos dentro de una subclase de Smalltalk son privados.

La vía iterativa DDE es manejada a través de sus dos subclases DDEClient y DDEServer.

Los siguientes son los miembros de la clase: DynamicDataExchange.

- DynamicDataExchange
- DDEClient
- DDEServer
- SmalltalkEmulator

La clase "DynamicDataExchange" contiene el ambiente privado y especifica los métodos a implementar en el mensaje del protocolo de DDE. En lugar de hacer una interface con la clase "DynamicDataExchange", el usuario hará una interface con las clases DDEClient o DDEServer. Las clases DDEClient o DDEServer utilizarán los métodos contenidos en ellas para proveer a la clase DynamicDataExchange de un nivel más alto de servicio y abstracción.

A continuación se presentan ejemplos de la forma como se utilizan las clases, para la creación de una aplicación servidor o cliente, incluyéndose el cómo se debe iniciar una conversación con otras aplicaciones.

En primer lugar las aplicaciones deben ya estar ejecutándose. Para comenzar una conversación como cliente, la aplicación que inicia envía el siguiente mensaje a la clase DDEClient.

```
DDEClient newClient: anObject application: applicationName topic: topicName
```

El objeto utilizado en esta declaración es decir "*anObject*" es un objeto de smalltalk que se desea sea el cliente, manejándolo como una variable. Application, applicationName topic, y topicName mantienen el mismo significado como ya ha sido explicado.

Una vez que una conversación DDE ha comenzado, el cliente puede usar el nuevo objeto "DDEClient" como un puente para interactuar con el servidor.

Para establecer un encadenamiento en caliente (hot link) con un ítem en el servidor, la aplicación cliente envía el siguiente mensaje al objeto DDEClient.

```
HotLinkItem: itemName class: aClass selector: aSelector.
```

"itemName" es una cadena que identifica el dato, "aClass" es un tipo de clase de smalltalk, y "aSelector" es un argumento doble que es llamado cuando el dato regresa del servidor al cliente.

De igual forma se maneja un mensaje para establecer un encadenamiento tibio (warm link) como se muestra a continuación.

WarmLinkItem: itemName class: aClass selector: aSelector

"itemName" es una cadena que identifica el dato, "aClass" es un tipo de clase de Smalltalk y "aSelector" es un argumento que es llamado cuando el dato es cambiado en el lado del servidor.

Un comando puede ser ejecutado en el servidor enviando el siguiente mensaje al objeto DDEClient.

ExecuteCommand: commandString

"commandString" es una cadena que contiene un comando que se ejecutará en el servidor.

REFERENCIAS DEL CAPITULO III

- Dan Shafer
Smalltalk Programming For Windows
1993 pp. 338-349
Prima Publishing
- Jeffrey D. Clark
Windows Programmer's Guide To OLE/DDE
1992 pp. 5-11
SAMS
- Thomas Chester
Excel 5 para Windows
1995
Ed. Ventura
- Tutorial and Programming Handbook
Smalltalk Programming
IBM Versión
- Artículos proporcionados por Microsoft.
1995-1996
- Microsoft Access "Creación de aplicaciones"
1994
Microsoft Corporation

IV

DOCUMENTACIÓN DEL SISTEMA

IV

DOCUMENTACIÓN DEL SISTEMA

IV.1 INTRODUCCIÓN

Para el desarrollo del sistema se utilizó el lenguaje SmalltalkV para Windows, con el objetivo de cumplir con dos propósitos, el primero contar con una forma sencilla de programar dado que Smalltalk es un lenguaje orientado a objetos, y cuenta con una extensa biblioteca de clases, además de que entre esas clases y los métodos existentes se encuentran los destinados para el intercambio dinámico de datos (Rutinas DDE), así como también una variedad de comandos que están destinados para el intercambio de datos. El segundo propósito, es facilitar al sistema el manejo de las rutinas DDE en su accionar sobre las aplicaciones. Cabe señalar que dado que el lenguaje de programación Smalltalk se encuentra en ambiente Windows, podrá ser fácilmente aprovechado por otras aplicaciones las cuales tengan a su vez la facilidad del manejo de las rutinas DDE, sumando esto a que estas rutinas formen parte de las librerías tanto de Excel como del propio lenguaje, y se encuentren en la sexta capa del modelo OSI (capa de presentación).

Otra de las grandes facilidades de Smalltalk, es su manejo de clases y métodos, en el sistema se crea una nueva clase, pero se utilizarán los métodos u objetos que ya se encuentran grabados en las clases de Smalltalk, es decir se maneja como un administrador de objetos, que sabe cuando utilizarlos y en que momento. Debido a esta cooperación de clases y objetos, la programación es más sencilla.

El sistema como se especificó anteriormente esta compuesto por dos aplicaciones, por lo que el sistema es modular, cuenta con dos módulos, uno de ellos es Excel y el otro Smalltalk, al primero de ellos se le llamará módulo de ejecución y al segundo, módulo de programación, cada uno por si solo es independiente, pero en este caso, ambos dependen de cada uno, si el contenido de Excel se modifica, Smalltalk inmediatamente modifica su funcionamiento. El módulo de programación se comprobará en el módulo de ejecución, cada instrucción al ejecutarse mandará una orden al módulo de ejecución para así crear el sistema de monitoreo.

En cuanto al mantenimiento del sistema se puede decir que si Excel se actualiza, como servidor, obligará al cliente a actualizarse de igual forma, el beneficio de este sistema es que si se tienen versiones posteriores de Excel en lugar de afectar al sistema de Smalltalk, lo beneficiará, pues es posible que algunas funciones que no se pudieron elaborar en la presente tesis, en un futuro esos problemas se pueden solucionar de diferente forma y sencillamente. Así como también, si Windows se actualiza, de igual forma mejorará los beneficios del sistema de monitoreo.

IV.2 DESCRIPCIÓN DEL PROGRAMA.

El programa se encuentra escrito en Smalltalk para Windows; DDETestClient será el nombre de la clase que lo identificará, dentro de esta clase se encuentran los métodos que al irse ejecutando permitirán que se haga el ligado de aplicaciones y comience el intercambio de datos y mensajes.

En primer lugar deberá procederse a hacer el ligado de Smalltalk con el primer Libro de Excel (Libro1), para ello tanto Excel como Smalltalk se deberán encontrar activos, ésto es que ambas aplicaciones deben ser abiertas por el usuario con anticipación para que pueda comenzar el ligado mediante la invocación del nombre del archivo de Excel desde Smalltalk, definiendo al primero como Servidor y al último como Cliente. Es decir que Smalltalk se

encargará de hacer las peticiones a Excel, y Excel de responder y enviar a Smalltalk los resultados de sus solicitudes o peticiones.

Ya que se ha establecido el ligado con Excel, usando el nombre del archivo deseado, Smalltalk abrirá el archivo solicitado a Excel el cual se monitoreará posteriormente utilizando la ejecución de comandos de Excel desde Smalltalk.

Una vez iniciada la conversación con un tópico definido, se indagará de los deseos del usuario, es decir, si él está de acuerdo en monitorear las fórmulas o simplemente el contenido de cada celda; posteriormente se tendrá que investigar cual es el tamaño de la hoja electrónica por monitorear, esto a fin de abarcar toda la hoja de cálculo, mediante comandos especiales de Excel cuya ejecución haya sido solicitada desde Smalltalk, ya que al iniciarse el Hotlink, este método se encarga de monitorear solamente las celdas indicadas, de otra forma, si se quiere observar la modificación de una celda que se encuentra fuera del rango a monitorear, esa celda simplemente no se monitoreará y no se sabrá si sufre algún cambio.

A continuación se da inicio a la fase de monitoreo de la hoja de Excel, para tal fin se hace un respaldo en forma automática de su contenido copiándolo en una "Matriz 1" la cual es del tamaño de la hoja que se va a monitorear obteniéndose así una imagen idéntica de los datos reales contenidos en la hoja por monitorear reflejando el estado inicial de la hoja. Posteriormente haciendo un encadenamiento en caliente (hotlink), se podrán obtener los datos que se hayan modificado en la hoja, permitiendo con esto que en cuanto un solo dato dentro de la página haya cambiado, Excel en forma automática manda un mensaje a Smalltalk anunciándole que la hoja en alguna de sus celdas ha sufrido un cambio, para a continuación mandar nuevamente todos los datos entre los cuales se encuentran las nuevas modificaciones y se almacenan en una "Matriz 2". Así sucesivamente todos los cambios efectuados por el usuario se almacenan en ésta matriz, hasta que el supervisor desee ver los cambios producidos por el usuario en la hoja.

En este momento se hace un análisis del contenido de cada celda para observar si tiene información numérica o de texto o si se encuentran fórmulas dentro de ella; se hace una comparación de matrices y de esta manera se podrá conocer que celdas han cambiado y en que consistió el cambio, esto es: si simplemente se cambió un dato o si su cambio produjo una modificación en otra celda debido a que alguna de las celdas involucradas en la fórmula haya cambiado, lo cual ocasiono que la celda correspondiente a la fórmula, se modificará. O bien cuando se están monitoreando las fórmulas de la hoja, podrá observarse si alguna de ellas se modifico, enviándose el mensaje correspondiente.

El sistema escrito en Smalltalk comenzará a mandar mensajes al usuario para indicarle cuales fueron las últimas modificaciones en su hoja. Esta actividad se hará cada vez que el usuario solicite observar las modificaciones, o simplemente desee que se contabilicen estas, o bien si desea continuar con el monitoreo. Las nuevas modificaciones se seguirán almacenando en la "Matriz 2" con los siguientes datos que hayan sido modificados para nuevamente comparar ambas matrices, y así saber cuales fueron los cambios sufridos e indicárselo al usuario, y así continuar hasta que el mismo usuario desee salir del sistema de monitoreo.

Finalmente si el usuario quiere salir del monitoreo, podrá hacer la petición a Smalltalk, a lo cual Smalltalk responderá preguntando al usuario si esta seguro de su decisión de abandonar el modo de monitoreo. Llegando a esta parte, la forma de cerrar la aplicación dependerá de la forma en que el usuario utilizó el sistema; si el usuario desea cerrar la aplicación y no ha ocurrido hasta ese momento un cambio, el sistema se cerrará pero la hoja electrónica activa de Excel permanecerá abierta, pues no se produjo ningún cambio. En dado caso de que si ocurra alguna modificación, el usuario deberá de tomar sus precauciones, pues primero se debe cerrar el sistema de monitoreo y después, Excel en forma automática al detectar el final de la ejecución preguntará a su vez al usuario si desea grabar los cambios sufridos en la hoja, y Excel cerrará su aplicación.

IV.3 DIAGRAMA DE FLUJO

A continuación se tiene el diagrama de flujo que representa la solución al problema que originó esta tesis, este es el procedimiento que se sigue para la elaboración del sistema, y se presenta en este capítulo a fin de tener una documentación formal del sistema que se elaboró para esta tesis.

El diagrama de flujo es la representación gráfica de lo que se desea que haga la computadora, por tal razón, es la parte principal de la planeación del sistema de esta tesis, y es la guía principal para su elaboración, ya que de esta forma se facilita la programación, pues se tiene una vista inicial de lo que se necesita, como variables y procesos a elaborar, es un gran auxiliar para el programador.

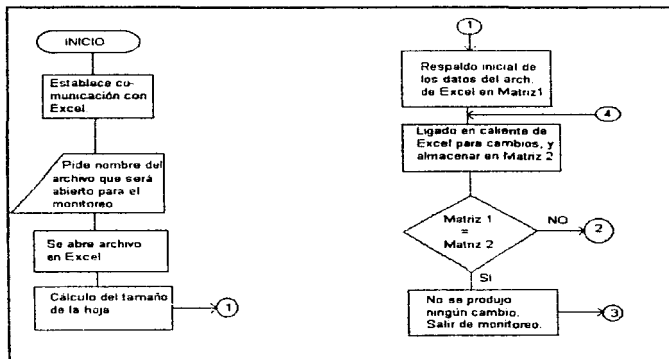


DIAGRAMA DEL SISTEMA PARTE I

GAP. IV. DOCUMENTACIÓN DEL SISTEMA

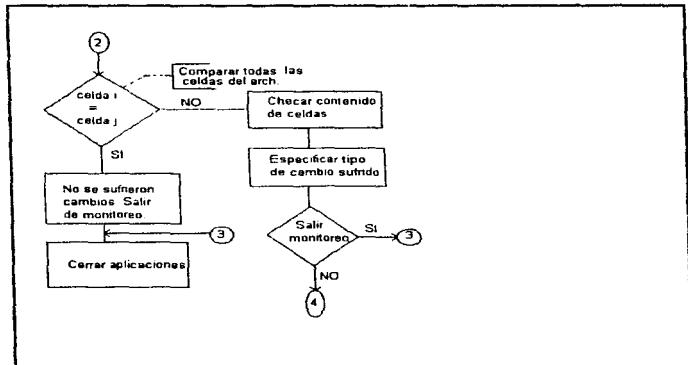


DIAGRAMA DEL SISTEMA PARTE II.

CAP. IV. DOCUMENTACIÓN DEL SISTEMA

IV. 4 LISTADO DEL PROGRAMA.

En el siguiente listado se muestra la definición de la clase principal, así como las clases que son utilizadas por el sistema, junto con los métodos que integran a cada clase. Se describe su funcionamiento, antes de cada párrafo.

A fin de clarificar al lector en el entendimiento de la lógica seguida en la programación del sistema de monitoreo, se adjunta a cada clase y método su descripción.

Programa principal en Smalltalk:

Se muestra la clase principal del sistema de monitoreo y las variables de instancia definidas por el programador, las cuales son utilizadas en todos los métodos que integran a esta clase:

```
TextWindow subclass: #DDETestClient
instanceVariableNames:
'dde arch mat l c n '
classVariableNames: "
poolDictionaries: "
WinConstants '
```

Los métodos que integran esta clase son:

DDETestClient class methods

Este es el método para hacer el ligado inicial con Excel, por default se liga con el Libro1 (hoja1 de Excel), posteriormente se envía un mensaje a ésta hoja, con la finalidad de mantener siempre abierto el Libro1, y así asegurar las ligas futuras con otros archivos. Smalltalk solicita el nombre del archivo que se desea monitorear; es posible observar en el texto del programa que es de suma importancia, especificar la ruta del archivo que será abierto. Smalltalk con el dato solicitado al usuario, abrirá el archivo, y terminará esta conversación inicial con Excel. Lo que significa:

CAP. IV. DOCUMENTACIÓN DEL SISTEMA

```
abrir
|alix|
  ( alix := DDEClient newClient: self application: 'Excel' topic: 'Libro1' ) isNil ifTrue:
  [MessageBox titled: 'Advertencia' withText: 'No puede inicializarse con el servidor'
  style: MbOk|MbIconexclamation.
  ^false]
  alix pokeItem: 'L4C3' object: 'BIENVENIDO A MONITOREO'.
  arch := Prompter prompt: 'De nombre de archivo en Excel para abrir,
  especificando directorio.' default: ''
  (alix executeCommand: ['Abrir"',arch,"'])
  ifFalse:[MessageBox message: 'El comando fue ejecutado exitosamente por
  Excel!].
  alix terminate.
```

Este método cierra la aplicación en el caso de que no hayan existido modificaciones en la hoja de Excel. Aquí se requiere aplicar un método especial, que consiste en finalizar la conversación sin cerrar la aplicación de Excel, pues de lo contrario se producirá un error. De esta forma simplemente se cerrará el sistema de monitoreo, sin que se comunique ningún mensaje a Excel. A continuación se muestra el método descrito:

```
cerrar
  (MessageBox confirm: 'Salir de monitoreo'
  ifTrue: [dde notNil ifTrue: [dde terminate]
  super close.
  ]
```

En caso de que hayan existido modificaciones en la hoja electrónica, se ejecuta el método siguiente. Si el lector compara el método anterior y este método podrá observar que la única diferencia que existe es que se manda ejecutar un comando de cerrar la hoja que se encuentra en monitoreo. Esto ocasiona que primero se tenga que cerrar la aplicación de Smalltalk y posteriormente la aplicación de Excel.

```
close
  (MessageBox confirm: 'Salir de monitoreo'
  ifTrue: [dde notNil ifTrue: [dde executeCommand: ['Cerrar'];
  terminate].
  super close.
  ]
```

CAP. IV. DOCUMENTACIÓN DEL SISTEMA

El siguiente método es complementario a los dos anteriores. Simplemente asegura que la ventana generada por el sistema de monitoreo será cerrada.

```
close: aPane
(MessageBox confirm: 'Salir de monitoreo')
ifTrue: [dde notNil ifTrue: [dde executeCommand: 'Cerrar']
        terminate]
super close
]
```

El siguiente método compara las dos matrices generadas para especificar que celdas sufrieron cambios, el lector puede observar que aquí se llama al método scan, que permite observar el tipo de cambio sufrido en las celdas.

```
compara
| i j cad1 cad2 stream1 stream2 s t |
i:=1 j:=1
[i <= l]
whileTrue: [
cad1:= mat atRow i atColumn j.
cad2:= n atRow i atColumn j
stream1:= WriteStream on: ".
i printOn: stream1.
s:='L', stream1 contents.
stream2 = WriteStream on: "
j printOn: stream2
t:='C', stream2 contents.
(cad1 equals: cad2)
ifFalse: [MessageBox notify: 'Mensaje para usuario' withText: 'La
celda ', s, t, ' ', 'sufrió cambios'.
self scan: cad1 to: cad2].
(j = c)
ifTrue: [j:=1.
i:=i+1]
ifFalse: [j:=j+1].
]
```

Dado que es de gran interés para el usuario conocer el número de cambios producidos en la hoja de Excel se elaboró un método dedicado a hacer esa actividad, este método es el que se muestra a continuación, el cual cuenta el número de modificaciones que se produjeron. Compara todas las celdas que integran las dos matrices; si no son iguales incrementa el contador, también especifica la posibilidad de que no se haya producido algún cambio, y especifica también el número de modificaciones que sufrió la hoja electrónica.

```

cuenta
{ i j cad1 cad2 a stream1 x}
a:=0 i:=1.
j:=1
if i <= l j
    while True: [
        cad1:= mat atRow i atColumn j.
        cad2:= n atRow i atColumn j.
        (cad1 equals: cad2)
            if False: [a = a+1].
        (j = c)
            if True: [ j:=1
                    i:=i+1]
            if False: [ j:=j+1].
        ].
x:= (l * c).
(x = a)
if True: [ MessageBox notify: 'Mensaje para usuario' withText: 'No hubo
modificaciones'.
a:=0
self cerrar].
stream1 := WriteStream on: ".
a printOn: stream1.
MessageBox notify: 'Mensaje para usuario' withText: '
stream1 contents,' ',
'modificaciones'.

```

A continuación se muestra el método dde, el cual genera una variable de instancia, que será utilizada para hacer el ligado con el archivo que ya se encuentra abierto.

```

dde
^dde.

```

CAP. IV. DOCUMENTACION DEL SISTEMA

Método de despliegue: solicita el número de celdas que tiene la hoja electrónica que fue abierta, se inicializan las dos matrices que serán llenadas con los datos de la hoja de Excel. En primer lugar se solicitan los datos iniciales para almacenarlos en la Matriz 1, después inicia el encadenamiento en caliente (hotlink) de las celdas especificadas al inicio del método y finalmente manda el mensaje de que se está monitoreando el archivo especificado. Hasta este momento es cuando comienza el monitoreo.

despliega

```
| string s t stream stream2 dato v1 i j |  
  
i:=1.  
j:=1.  
c := (Prompter prompt: 'De cuantas columnas consta la hoja ' default: ") asInteger  
l := (Prompter prompt: 'De cuantos renglones consta la hoja ' default: ") asInteger.  
mat:= (Matriz rows: l columns: c)  
n:= (Matriz rows: l columns: c).  
  [ i <= l ]  
  while True: [  
    j:=1.  
    stream := WriteStream on: ".  
    i printOn: stream  
    s:=L1, stream contents.  
    [ j <= c ]  
    while True: [ stream2 := WriteStream on: ".  
      j printOn: stream2.  
      t:='C'; stream2 contents.  
      string := s.t  
      dato := dde requestItem: string class: String.  
      mat atRow: i atColumn: j putData: dato.  
      dde hotLinkItem: string class: String selector:#item:string.  
      j := j+1  
    ]  
    i := i+1  
  ].  
  
self show: ' ', self cr.  
self show: ' ', self cr, self cr, self cr.  
self show: ' Se está monitoreando el archivo: ', arch.
```

CAP. IV. DOCUMENTACIÓN DEL SISTEMA

El siguiente método pregunta si el usuario desea monitorear las celdas que contienen fórmulas en la hoja electrónica, si es así ejecuta una macro que ha sido previamente grabada en Excel por el supervisor y se encuentra en el libro de macros: "Personal.xls".

```
formu
[new]
(MessageBox confirm: 'Desea monitorear las fórmulas?')
if True: [new := DDEClient newClient: self application: 'Excel' topic: arch
         (new executeCommand: '[Ejecutar]"PERSONAL.XLS!formulas"]')
         new terminate.]
```

En respuesta a este método Smalltalk inicializa la sesión DDE con el archivo que fue especificado por el usuario.

```
initiate: application topic: topic

(dde := DDEClient newClient: self
 application: application
 topic: topic) isNil if True:
[
  MessageBox titled: 'warning' withText: 'Couldn't Initiate with server' style:
  MboOk|MbiConexclamation.
  ^False ]
```

Este método fue utilizado durante los tipos de pruebas que se aplicaron al sistema, el cual es de suma importancia ya que las modificaciones se pueden observar en la pantalla, y en cuanto sucede un cambio. El hotlink invoca este método para que se observe la nueva modificación.

```
item: item string2: string

"The server has updated a hot linked item ."

self show:item; show:' ' ; show:string.
```

CAP. IV. DOCUMENTACIÓN DEL SISTEMA

El siguiente método se encarga de actualizar los datos que se obtienen del encadenamiento hotlink, y al recibir las nuevas modificaciones las almacena en la Matriz número 2:

```
item item string string
  |ren col|
col:={{(item reversed upTo: $C) reversed) asInteger}
ren:={{(item upTo: $C) copyWithout: $L} asInteger}
n atRow: ren atColumn: col putData: string.
```

Este es el método que se encarga de abrir todo el sistema de monitoreo, éste es el método inicial, que va llamando a todos los métodos que integran la clase DDETestClient de Smalltalk para que se ejecuten en un orden especificado:

```
openExcel

super openOn: '. '
self mainView when: #close perform: #close..
self abnr:
  formu:
  initiate: 'Excel' topic: arch;
  despliega.
```

Se elabora un scanner, es decir se utiliza un método de búsqueda para conocer la causa por la que la celda sufrió un cambio, y comienza a enviar los mensaje indicados al usuario:

```
scan: cad1 to: cad2

| ele1 ele2 men |
ele1 = cad1 first. ele2 = cad2 first.
(ele1 ~= $= and: [ele2 ~= $=])
  ifTrue: [MessageBox notify: 'Mensaje para usuario' withText: 'El contenido de
la celda ha cambiado].
(ele1 = $= and: [ele2 ~= $=])
  ifTrue: [MessageBox notify: 'Mensaje para usuario' withText: 'Se tenía una
fórmula, se ha cambiado por una fórmula].
(ele1 ~= $= and: [ele2 = $=])
  ifTrue: [MessageBox notify: 'Mensaje para usuario' withText: 'Se tenía un texto,
se ha cambiado por una fórmula].
(ele1 = $= and: [ele2 = $=])
  ifTrue: [MessageBox notify: 'Mensaje para usuario' withText: 'La fórmula ha
cambiado].
```


CAP. IV. DOCUMENTACIÓN DEL SISTEMA.

Como podrá observar el lector los métodos no han sido presentados en el orden en que ejecutan en la realidad, simplemente se encuentran organizados en forma alfabética y en orden ascendente, pero podrá distinguir que el método encargado de administrar los métodos es el de openExcel.

Con la descripción anterior se da por cumplido el requerimiento de presentar la clase y métodos que conforman el sistema de monitoreo, sin embargo será de mayor claridad para el lector el acceder a ellas a través del diskette que acompaña esta tesis. Las clases, objetos y sus métodos se encuentran en el archivo Monitoreo.cls, el lector interesado en efectuar ejecuciones de prueba del sistema podrá constatar que las bibliotecas no incluidas en el diskette serán complementadas en el momento de la compilación por el compilador de Smalltalk.

REFERENCIAS DEL CAPITULO IV.

- Dan Shafer
Smalltalk Programming for Windows
Prima Publishing
- Thomas Chester
Excel 5 para Windows
Ventura Editores
- Tutorial and Programming book
Smalltalk/V for Windows.
IBM Versión
- Direcciones utilizadas para la consulta de información en Internet.
<http://www.andyone.on.ca>
<http://www.microsoft.com>



PRUEBAS DEL SISTEMA



PRUEBAS DEL SISTEMA

V.1 METODOLOGÍA PARA LAS PRUEBAS.

Para llevar a cabo el desarrollo y las pruebas del sistema, se requirió contar con la siguiente plataforma: un equipo que consta de una PC en donde se encuentra cargada la última versión de Excel en nuestro caso la 5.0, también el compilador del lenguaje orientado a objetos Smalltalk/V 2.0 para Windows y la plataforma básica para el manejo de ambas aplicaciones la cual es Windows.

Se necesita un mínimo de 4 Megas en RAM, y un procesador 486. De la velocidad del procesador dependerá la rapidez con que se ejecute el sistema, así como también del ciclo de acceso a memoria. Consideramos importante señalar que las rutinas DDE así como OLE se pueden manejar casi sin problemas estando bajo el sistema operativo Windows de cualquier versión. También es importante hacer notar que no es necesario tener el compilador y la plataforma de Smalltalk presente, una vez compilado el sistema y teniendo el ejecutable, simplemente se requiere que el Excel 5.0 se encuentre residente.

El propósito central de esta tesis fue el diseño y construcción del sistema de monitoreo. Así sus pruebas fueron planeadas para realizarse en diversas etapas: al inicio solo se observaría si los comandos entre aplicaciones se ejecutaban de la forma adecuada, para posteriormente agregar las hojas de cálculo y ejecutar los comandos necesarios para observar si era posible

accesar sus datos y si las hojas no perdían su funcionalidad comportándose de la manera en que se esperaba.

V.2 PRUEBAS.

Se elaboró un conjunto de hojas electrónicas en Excel, con las cuales se probaría el ligado de aplicaciones, así como todas las pruebas concernientes con el sistema. En dichas hojas se colocarían una serie de funciones en celdas seleccionadas estratégicamente a fin de poder emular la movilidad que cualquier usuario pudiera ejecutar sobre una hoja determinada.

A fin de establecer un procedimiento riguroso en cada paso que incluyera una modificación a una celda, la entrada de un comando, o el establecimiento de una condición particular, se supervisaría si en realidad el sistema respondía en forma correcta de acuerdo a lo que se esperaba, sobre todo en lo referente a el ligado con la aplicación y la ejecución de comandos desde Excel hacia Smalltalk, servidor y cliente respectivamente, comprobando en cada ocasión que cada uno de los componentes modificara su estado e hiciera lo debido. Esto provocó algunos contratiempos, sobre todo porque habría que saber la sintaxis exacta para la ejecución de los comandos, mediante las rutinas DDE. Cabe aquí hacer mención que parte de las pruebas realizadas se establecieron a manera de ensayo-error, dado que en la literatura obtenida de diferentes fuentes no se incluye el formato exacto que se debe cumplir en el traslado de información, de una aplicación a otra.

Al hacer las pruebas pertinentes al sistema, se pudo observar que no se obtenían los resultados apropiados fácilmente, ya que en varios de los casos estudiados referentes al ligado entre Smalltalk y Excel, la sintaxis utilizada para la programación de los métodos encargados de lograr el éxito del ligado entre las aplicaciones, estaba mal manejada, por ejemplo: cuando se quería ejecutar una macro ya elaborada, se pretendía utilizar el comando de ejecución de una macro en inglés y seguir la sintaxis planteada en este idioma, pero el Excel que

se maneja está en español, y la sintaxis del comando de ejecución en Smalltalk (siguiendo la información obtenida en la literatura), se encontraba errónea, esto provocó un atraso en el módulo de programación correspondiente, sin embargo después de probar varios comandos y de investigarlos intensamente, se pudo encontrar la sintaxis correcta.

Otra prueba realizada fue para ver si en realidad se podía iniciar el ligado con la hoja correcta utilizando para tal efecto el nombre de la hoja, para lo cual se envía el comando de apertura de hoja, una vez abierta esta hoja de trabajo, se observa el resultado, que es la propia apertura de la hoja en cuestión. Se hace hincapié en la importancia que tiene la apertura de la hoja inicial, dado que es el eslabón que será utilizado como puente durante toda la actividad del usuario.

Una vez iniciado el ligado con esa hoja, se solicitan ciertos datos, que se conoce con seguridad existen dentro de la hoja, observando si efectivamente los datos a monitorear son exactamente los que ya habían sido capturados con anterioridad, de esta manera se comprobó el ligado con la hoja en forma satisfactoria.

A fin de realizar una prueba referente a el acceso de datos en la hoja abierta y ligada, se crearon diversos módulos de programas que permitían observar la modificación de una celda determinada de acuerdo al siguiente procedimiento, primero se modifico una celda que no repercutiera en otra, verificando de inmediato que la hoja no hubiera sufrido cambio alguno en sus componentes, debido a la modificación así realizada, registrando automáticamente la existencia de cualquier cambio o modificación. Para esta fase se hizo uso del encadenamiento hotlink, el cual represento una ayuda considerable.

Inicialmente esta actividad se observaba en el monitor, si alguna celda cambiaba, inmediatamente aparecía el cambio en el monitor, pero se presentaban todas las celdas que se encontraban en monitoreo, junto con la

CAP. V. PRUEBAS DEL SISTEMA.

celda que acababa de modificarse, lo cual impedía a golpe de vista, discriminar en donde se había realizado el cambio. Para facilitar tal problema fue necesario desarrollar un programa capaz de manejar una matriz cuyo contenido fuera una copia inicial de los datos encontrados en una hoja completa, con lo que recorriendo y comparando celda por celda, la página origen contra su imagen, se hizo posible el discriminar las modificaciones en forma automática, dejando como una ayuda importante en el menú del supervisor el poder contabilizar y visualizar estas modificaciones. De esta manera, en el momento en que el usuario lo desee, puede observar esas modificaciones.

A continuación se procedió a introducir en la hoja de prueba, celdas que pudieran modificar a otras o ser modificadas en forma indirecta de tal manera que se cubrirían los casos en los que el usuario pudiera incluir un proceso iterativo de modificación entre celdas. En este caso, y siguiendo el procedimiento anterior, se visualizó la problemática encontrada en la contabilidad de las modificaciones, ya que era posible perder algunos cambios que se hubieran dado debido al cambio o frecuencia de una celda, una o más veces seguidas. El sistema avisaba con oportunidad cuando se habían tenido dos modificaciones, o bien cuando la celda se había modificado desde el teclado, y aún más, la celda que se modificaba como producto del último cambio ocurrido.

Después de observar que todas las modificaciones se contabilizaban en forma correcta, se inició la elaboración del módulo que observa y supervisa estas modificaciones, lo cual se hizo utilizando la misma aplicación de Excel, la cual devuelve como parte del ciclo, la localización de la celda que se modifica. A continuación y como parte inicial de las pruebas se incluyó en el programa la facilidad de poder imprimir tal dirección colocándola en forma entendible para el usuario. A fin de analizar internamente el resultado del monitoreo se observó detenidamente la forma como podría encontrarse y observarse la modificación que había ocurrido.

Para probar el monitoreo de fórmulas, se tuvo que desarrollar un módulo de programación aparte, y preguntar al usuario si tenía algún significado en particular el visualizar si se habían modificado o cambiado las fórmulas, pues se tiene el inconveniente de que o se monitorea el contenido de las celdas de una página o pasando a la página de fórmulas se monitorean éstas.

Para este nuevo tipo de monitoreo se hace un ligado especial, pues se inicia el monitoreo con el archivo que contiene las fórmulas, en lugar del contenido de las celdas, accionándolo mediante una macro que se ejecuta en el momento en que el usuario especifica que desea monitorear las fórmulas, de tal manera que puede observarse si se cambió la fórmula y de nueva cuenta la búsqueda y discriminación de donde se llevó a cabo tal cambio, obviamente la labor de revisar celda por celda en una página de fórmulas no es una labor sencilla, debido a que cada signo es importante dentro de un conjunto de patrones específicos.

Finalmente fue necesario desarrollar un procedimiento para llevar a cabo la despedida del sistema, lo cual incluiría el cierre de las páginas de Excel y su respectiva despedida. Para tal fin, se decidió que de existir cambios en la hoja, el usuario sería encargado de salvar o no el reporte de las modificaciones o cambios producidos en la hoja de Excel, para posteriormente salir del sistema de monitoreo, esto se logró utilizando las señales enviadas por Excel a Smalltalk, a fin de que este último se enterase de que Excel estaba cerrándose. Uno de los casos estudiados que debieron de ser tomados en cuenta por nuestra aplicación fue el considerar que sucedería si la hoja de Excel no hubiera tenido cambios, ya que Excel no enviaba ninguna señal, y en consecuencia no se cerraba y el sistema marcaba un error. Para resolver este problema, se tuvo que elaborar un nuevo método de cerrado, en donde el sistema se cierra sin necesidad de pedir autorización a Excel, y su hoja se queda abierta.

VI

RESULTADOS

VI

RESULTADOS

VI.1 SOLUCIONES AL PROBLEMA PLANTEADO.

En esta sección se resumen las soluciones aplicadas en el sistema de monitoreo para la resolución de los problemas que operativamente se le presentarían al supervisor durante el monitoreo en el manejo de una hoja electrónica de Excel.

Dado que el procedimiento aplicado durante las pruebas para poder detectar cualquier tipo de error o desviación que se hubiera podido presentar desde la apertura hasta la cerradura de la hoja, fue exitoso, se pudo detectar y solucionar puntualmente el conjunto de problemas que se presentaron durante el desarrollo, y no se utilizó otro procedimiento de pruebas para el monitoreo de las diversas celdas.

Las soluciones aplicadas fueron particularmente: la utilización del encadenamiento en caliente (hotlink), pudiéndose observar cualquier cambio dentro de la hoja electrónica al vuelo, reduciéndose nuestro problema a capturarlo en la misma forma y hacer el reporte necesario a fin de no perder tal información, ya que si el usuario hiciera una sola modificación, y fuera pasada por alto en el sistema de monitoreo, todo el sistema quedaría fuera de uso ipso factum.

CAP. VI RESULTADOS

En el caso de cálculos iterativos y en particular de la hoja de fórmulas se decidió que el sistema conmutara las hojas supervisadas a la hoja de fórmulas con lo que el problema se reduce a tener que copiar y revisar el contenido de la copia de la hoja de fórmulas, transportando el problema original a conocer con precisión cual sería el tamaño de la hoja original ya que el sistema construido en la forma inicialmente planteada lo podría indicar sin dificultad señalándole al supervisor si existió un cambio en la fórmula, además de que si lo deseaba podría mandar imprimir la hoja de fórmulas. Cabe aquí hacer la siguiente aclaración, el sistema diseñado es capaz de monitorear una cosa a la vez, el contenido de cada celda, o las fórmulas contenidas en cada una de ellas.

La decisión de monitorear contenido o fórmulas dependerá del supervisor, el beneficio que se obtendrá con este sistema al monitorear el contenido de celdas es que si se modifica alguna celda y tiene repercusión en otra, todos los cambios producidos se avisaran al usuario del sistema, en este caso el supervisor; en cuanto al monitoreo de fórmulas, solo se podrá observar el cambio de la fórmula o alguna modificación en el contenido de una celda, pero si ésta repercute en la celda que contiene la fórmula, no se podrá observar el cambio, porque solamente se desea saber si hay modificación en la fórmula, y la fórmula no ha cambiado, simplemente una de las celdas de referencia modifico su valor.

Otro problema descrito fue el de cambiar los valores de las celdas accidentalmente. En este sistema no pueden existir accidentes, pues en cuanto se produzca una modificación cualquiera, se reportará al supervisor.

El monitorear las fórmulas también tiene su beneficio pues en cuanto el usuario desee observarlas lo podrá hacer, y se dará cuenta si el reporte estuvo bien hecho, incluyendo fórmulas, o fue simplemente una copia de algún reporte sin incluir fórmula alguna. El supervisor al especificar el monitoreo de fórmulas podrá observar si concuerdan las cantidades, o se cometió el error de no utilizar fórmula para los cálculos. Además de que si se desea se puede grabar

CAP. VI. RESULTADOS

el archivo con fórmulas, o mandarlo imprimir con ellas y posteriormente mandar imprimir el contenido, así podrá observar si columnas, renglones y totales checan con lo estipulado o lo esperado.

Otro problema descrito fue el copiado de celdas en otros procesos de cálculo dentro de la misma hoja, éste problema se soluciona de igual forma, al anterior, ya que de tener una celda como referencia en un valor de otra celda que se encuentra monitoreada y a la vez si la celda de la que nos ocupamos se encuentra también monitoreada, los cambios producidos en ambas se avisaran al usuario.

Se podrá observar que si todas las modificaciones repercuten en otras celdas, y se encuentran en la fase de monitoreo, inmediatamente se indicará su modificación y el motivo. De igual forma sucede con las fórmulas.

Cabe aclarar que en este trabajo se esperaba poder monitorear todas las hojas que formaban parte de un libro en una hoja electrónica de Excel, sin embargo el ligado con la aplicación por default se hace solamente con la hoja 1 del libro en cuestión, así que si se desea monitorear otra hoja se podrá hacer siguiendo los pasos que se le indican al supervisor en el apéndice del manual del usuario del sistema.

Salvo la especificación anterior, a la cual ya se dio una solución, se tuvieron otros contratiempos que se fueron superando mediante la investigación, estos contratiempos tuvieron que ver con la programación de comandos de Excel desde Smalltalk, ya que estos comandos deberían de ser invocados desde Smalltalk y ejecutarse en Excel. Prueba de ello, es la necesidad de la ejecución de una macro que permita pasar el contenido de las celdas a fórmulas, y de otro comando para mandar un mensaje escrito a un libro seleccionado y también el de abrir y cerrar la aplicación.

VI.2 RESULTADOS GENERALES.

Uno de los contratiempos que se tuvo fue en el momento en que se pretendió obtener el tamaño de la hoja de cálculo, pues Excel no tiene una función o comando que permita obtener este dato, así que se pensó en la elaboración de un pequeño programa que obtuviera el tamaño de la hoja, pero era demasiado aleatorio y con la posibilidad de que el dato que se obtuviese fuera erróneo, así que se optó por hacer que el usuario participara en este momento y otorgará el tamaño de la hoja.

En cuanto a las pruebas para la apertura y cerrado del sistema de la hoja electrónica, ésto fue una actividad sencilla, aunque se presento el problema de que al no existir cambios en la hoja electrónica de Excel, el sistema de monitoreo generaba un error, el cual consistía en que: si se deseaba salir del sistema de monitoreo, el usuario recibía automáticamente un mensaje de error, indicándole que la aplicación de monitoreo no podía cerrarse. Para resolverlo, se elaboró un nuevo método para cerrar la aplicación y que inhibiera el error de la aplicación del sistema; en este caso Excel no cierra su hoja electrónica automáticamente, simplemente se cierra el Sistema de Monitoreo, ésto debido a que no se tiene ninguna modificación en Excel.

El lector se puede dar cuenta, de que las rutinas DDE no se ejecutaron desde la primera vez pues hubo contratiempos, pero después de resolver el problema descrito y ejecutar las rutinas, se obtuvieron los resultados que en realidad se deseaban.

Se puede decir que la programación de las rutinas DDE es sencilla, pero al combinarlas con un lenguaje de programación orientado a objetos (Smalltalk), se complican, pues de cierta manera, se tienen que combinar las ordenes y esperar que la instrucción sea la correcta y se ejecute exitosamente, de otra manera se tiene que continuar verificando su sintaxis para su ejecución,

CAP. VI. RESULTADOS

ya que el problema se confunde con falta de sincronización entre los eventos y su atención.

El resultado general fue que para solucionar los problemas encontrados se tuvo que hacer una investigación ardua sobre la sintaxis correcta de las instrucciones que integraban un método, el cual influía directamente en el ligado de las aplicaciones debido a que no se podía comenzar el intercambio de datos.

No fue una labor sencilla, pues no existe un libro que explique paso a paso la programación de los métodos de Smalltalk para el ligado de aplicaciones, se tuvo que checar la programación de los métodos existentes y averiguar qué parámetros eran los adecuados para obtener los resultados requeridos. Todo ello mediante el sistema de prueba y error, pues se programaba de una forma y si no se obtenía el resultado necesario, se hacía una nueva programación y así sucesivamente hasta que se logró el objetivo de ligar las aplicaciones de forma adecuada, para así, comenzar el intercambio de datos entre Smalltalk y Excel.

Por su parte la información interna de este producto de Microsoft, también fue investigada a profundidad, encontrando que aún la información más simple, es decir la de la sintaxis de los comandos, parecería que no fue dada a la luz por los diseñadores, siendo uno de los secretos comerciales de mayor interés para el intercambio de datos entre aplicaciones de diferentes proveedores.

CONCLUSIONES

CONCLUSIONES

INTRODUCCIÓN.

Este capítulo incluye los logros que se obtuvieron al elaborar esta tesis, y el alcance que tendrá la aplicación del sistema dentro de una empresa, al obtener un trabajo de mayor calidad.

Pudieron observarse las ventajas del sistema al aplicarlo a un reporte contable, haciéndolo extensivo a cualquier tipo de reporte que se requiera en la práctica.

Dado que el objetivo principal de esta tesis fue: monitorear actividades realizadas sobre una o más hojas de cálculo de Excel ligadas con Smalltalk usando las facilidades de DDE, se puede especificar claramente que se ha cumplido con el objetivo, pues la elaboración del sistema de monitoreo se basó completamente en las rutinas de DDE, sin éstas no se habría solucionado el problema, ya que estas rutinas son la base principal del sistema.

Mediante ellas se hizo el ligado inicial con la aplicación además se tomaron los datos iniciales de la hoja electrónica por monitorear y los datos actualizados mediante un encadenamiento en caliente (hotlink). Finalmente se llevo al cabo el monitoreo de fórmulas. Se abrió y se cerro la aplicación de Excel.

Cualquier tipo de actividad hecha dentro de la hoja que se encuentre en monitoreo, fue especificada al usuario, esto es que cualquier cambio que se produjera por pequeño que este fuera, le sería comunicado.

CONCLUSIONES

Un sistema de monitoreo como el presentado es de suma importancia dentro de las actividades de una empresa, pues si se quiere proteger un documento, puede hacerse simplemente monitoreándolo. Si este sufre cambios, le serán indicados al usuario.

Al especificar el monitoreo de fórmulas, podrá observar si el reporte contiene las fórmulas especificadas y necesarias para que el reporte concuerde con los hechos contables o financieros que se esperan, así como el chequeo de totales de columnas o renglones.

En cuanto a la elaboración del sistema cabe hacer notar, que el lenguaje de Smalltalk facilitó la programación de las rutinas de DDE, el ligado entre aplicaciones y sobre todo el intercambio de datos. Todo ello mediante las clases y objetos que integran a este lenguaje que permite además que se maneje como un administrador de computadoras, y que especifica en que momento debe entrar a trabajar con determinado sistema.

Esto es que para la programación de Smalltalk se debe hacer una clase con sus objetos que indiquen en que momento específico y en que orden se deben ejecutar, lo cual se hizo al diseñar un grupo de objetos orientados en rutinas DDE, las cuales se ejecutaron en cierto orden, lo que permitió hacer realidad el monitoreo de datos.

Se especificó que existían tres tipos de monitoreo, pero el encadenamiento seleccionado para la elaboración del sistema resultó ser el mejor de los tres, debido a que en el momento en que ocurre la modificación, está es inmediatamente transferida a Smalltalk, donde se hacen los movimientos necesarios para saber que tipo de modificación ha sufrido la celda o las celdas.

CONCLUSIONES

Finalmente se logró el objetivo planteado, pues se obtuvo el sistema en sí, que hace lo que inicialmente se propuso. El tema tratado de intercambio dinámico de datos es actual, sin particular cuando se maneja con un lenguaje orientado a objetos.

Esta tesis integra un tema nuevo, que incluye la solicitud de datos de una a otra aplicación haciéndolo en forma dinámica y no solo mediante el cortado y copiado de datos entre aplicaciones.

En cuanto a tecnología es importante el haber trabajado en una aplicación cliente/servidor.

Estoy segura que este tema servirá como base para un trabajo futuro que permita hacer modificaciones y aporte más soluciones al usuario.



APÉNDICE:

MANUAL DE USUARIO



MANUAL DE USUARIO

Bienvenido al sistema de monitoreo, al utilizar este sistema trate de seguir las indicaciones en la forma propuesta pues de lo contrario podría producir un error de la aplicación de Smalltalk, se ha tratado de facilitarle de la mejor manera su manejo, el sistema esta integrado por dos aplicaciones, como es lógico si falla una aplicación también fallará la segunda.

Como primera recomendación, copie la macro "Personal.xls", que se encuentra en el disco, en el libro de macros de Excel. Después continúe con los siguientes pasos:

1. Inicialmente se debe tener la aplicación de Excel activa, es decir abrir Excel.
2. Minimizar su ventana para poder observar los cambios ocurridos.
3. Si el usuario desea que no se den cuenta de que el sistema de monitoreo está activo se puede minimizar la aplicación del sistema y maximizar la aplicación de Excel.
4. Una vez abierto Excel, se debe asegurar que se tiene el libro 1 como primer hoja de cálculo, Excel abre por default ese primer libro y el ligado se hace exactamente con ese libro. En caso de que no encuentre este libro 1 como primera hoja, vuelva a iniciar la aplicación de Excel.
5. Llame al sistema de monitoreo mediante el programa V.EXE debiendo aparecer en su pantalla, la plantilla que se muestra en la figura 9.

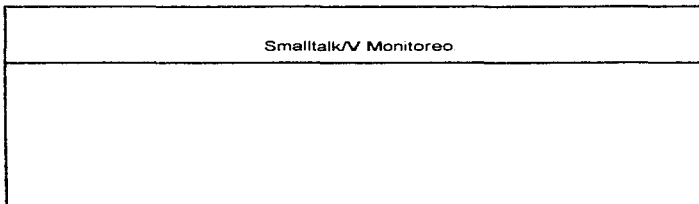


FIGURA 9. PANTALLA DEL SISTEMA DE MONITOREO

Durante la apertura y el ligado de la aplicación, aparecerán mensajes de confirmación, estos mensajes son manejados por la ejecución de la rutina Timeout, que determina la cantidad de tiempo que un control (cliente) espera para establecer la conversación con el servidor. Se mide en decimas de segundo y el valor por defecto es 50; si usted desea seguir esperando, se da Yes, se recomienda escoger No, y continuar con el sistema. Debe aclararse que los comandos tienen un tiempo determinado para ejecutarse, si el tiempo termina, se le solicita al usuario que seleccione si continua esperando o no.

El sistema de monitoreo pedirá el nombre del archivo a monitorear, si se encuentra en un directorio específico, debe indicarlo, ya que se encuentra abierto el archivo en Excel. Si desea monitorear una hoja distinta a la que se encuentra en ese momento, se debe hacer el siguiente movimiento:

Si tengo la hoja1 y deseo monitorear la hoja2, se debe pasar a la aplicación de Excel y mover la pestaña de la hoja2 al lugar de la hoja1, estas pestañas se encuentran en la parte inferior de la hoja, como se muestra en la figura 10.

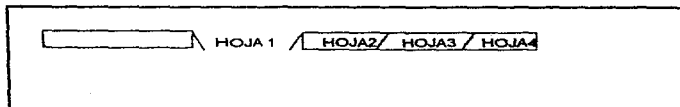


FIGURA 10. PESTAÑAS DE INDICACIÓN DE HOJAS EN EXCEL.

Si se desea monitorear la Hoja2, se cambia de lugar con la Hoja1, como se muestra en la figura 11.

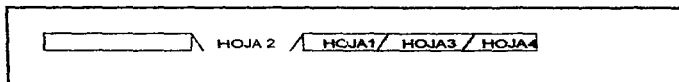


FIGURA 11: INTERCAMBIO DE HOJAS INDICADO POR LAS PESTAÑAS.

Si es otra hoja diferente la que se desea monitorear, simplemente se cambia de lugar con la hoja que se encuentra en el primer lugar, o que se encuentra activa.

Debe quedar claro que este procedimiento se hace en cuanto se abre el archivo y antes de contestar al mensaje de si desea monitorear las fórmulas.

Si desea monitorear la hoja que en ese momento está activa, continúe con el procedimiento, que es el de contestar la pregunta de si desea monitorear las fórmulas, aquí se debe estar seguro de si se monitoreará el contenido de las celdas o las fórmulas contenidas en las celdas, porque si se escoge el monitorear las fórmulas y después se quiere monitorear el contenido de las celdas, se tendrá que volver a iniciar el sistema de monitoreo.

Después de que se tomó la decisión, se pregunta de cuantas columnas y renglones está integrada la hoja. En esta parte conteste con exactitud. Si no llega a cubrir todas las celdas, las que quedaron sin especificar, no repercutirá su cambio en las otras celdas que si se encuentran en monitoreo.

En cuanto aparezca el mensaje de que se está monitoreando el archivo determinado, se puede decir que ha comenzado el monitoreo. Esto se muestra en la figura 12, donde se observa que la pantalla indica que está lista para el monitoreo del archivo.

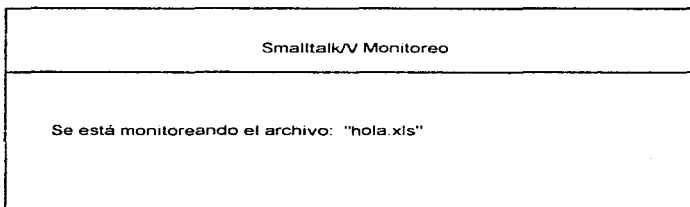


FIGURA 12: PANTALLA DE INDICACIÓN DE INICIO DE MONITOREO

Y el sistema está listo para reportar al usuario cualquier modificación que haya ocurrido, para que el usuario logre ésto, se deberá oprimir el botón derecho del mouse y aparecerá el menú que se muestra en la figura 13.

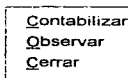


FIGURA 13: MENÚ DE MONITOREO

APÉNDICE. MANUAL DE USUARIO

Contabilizar: es para ver cuantas modificaciones han ocurrido hasta el momento.

Observar: es para ver qué tipo de modificaciones sucedieron últimamente.

Cerrar: es para salir de la aplicación.

En dado caso de que no haya ocurrido ninguna modificación y elige Contabilizar, aparecerá un mensaje de que No hubo modificaciones. Y se pregunta inmediatamente si quiere salir de monitoreo, se recomienda que si no va a hacer alguna modificación, salga de esta forma, porque debido a que no existió tal modificación Smalltalk al tratar de cerrar la aplicación de Excel no obtendrá alguna señal que le indique que Excel se cerrará y el Sistema de Monitoreo producirá un error. Es por ello que se recomienda que salga de esta forma. Si quiere observar las modificaciones cuando no hay tales, se producirá un error en la aplicación del Sistema de Monitoreo.

En dado caso de que ocurra una modificación, la podrá Contabilizar, Observar y sobre todo salir del sistema mediante la orden de Cerrar que se encuentra en el menú.

Cuando el usuario ha decidido salir del monitoreo y existieron cambios en las celdas, deberá esperar a que aparezca un mensaje de confirmación del Sistema, para que se cierre el Sistema de Monitoreo y a continuación se tendrá un mensaje de alerta en Excel, en este paso la aplicación de Excel comienza a parpadear, éste es debido a que Excel está generando un mensaje para el usuario y necesita la confirmación de éste. El usuario debe tomar la decisión que le atañe en ese momento, pues en cuanto le conteste el mensaje, se cerrará el archivo de Excel que fue abierto. Y aparecerá en el Libro 1 el mensaje de: Bienvenido a Monitoreo.

APÉNDICE. MANUAL DE USUARIO

Este mensaje le indica al usuario que está listo para que se vuelva a ejecutar el Sistema de Monitoreo, en el momento en que él lo desee.

En dado caso de que el usuario quiera quitar el formato de fórmulas, tiene que salir de Monitoreo, y en Excel ejecutar la macro "regform.xls", esto lo hace dirigiéndose al menú de Herramientas de Excel, escoger Macro y aparecerán todas las macros que se encuentran disponibles, y se escoge la macro indicada y se ejecuta.

Su archivo volverá a tener solo texto en las celdas, sin ninguna fórmula, y podrá guardarlo o imprimirlo en el formato deseado, con fórmulas o sin ellas.