

**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

LOTUS 1-2-3 AVANZADO

MATERIAL DIDACTICO

MAYO - JUNIO, 1992.

1-2-3 DE LOTUS 3.0 AVANZADO

OBJETIVO:

El alumno al final del curso aplicará un conjunto de instrucciones para el manejo y análisis de la información conocerá y aplicará las instrucciones que permiten programar en 1-2-3 versión 3.0 lo que llevará al alumno a poder optimizar sus procesos semi-automatizados, convirtiéndolos en automatizados, generar nuevos procesos de mayor complejidad y obtener el máximo provecho de 1-2-3.

DURACION: 30 Horas.

TEMARIO

I.- MANEJO DE INFORMACION

- I.1.- Manejo de Tablas
- I.2.- Estadísticas con la Base de Datos
- I.3.- Resumen de Información

II.- MANEJO DE MACROS

- II.1.- Importancia
- II.2.- Estructura
- II.3.- Teclas Especiales
- II.4.- Creación
- II.5.- Ejecución
- II.6.- Ejemplo Práctico

III.- LENGUAJE DE PROGRAMACION DE 1-2-3

- III.1.- Conceptos Básicos
- III.2.- Manejo de Datos
- III.3.- Lectura de Información del Usuario
- III.4.- Manejo del Flujo de Control de la Macro
- III.5.- Uso de Subrutinas
- III.6.- Manejo de Menús
- III.7.- Control en el Despliegue de la Pantalla
- III.8.- Manejo de Errores
- III.9.- Procesamiento de Textos y Archivos Secuenciales
- III.10.- Otras Macros

IV.- APLICACIONES

- IV.1.- Rutinas Típicas
- IV.2.- Tips
- IV.3.- Ejemplo Práctico
 - Especificación
 - Diseño
 - Programación
- IV.4.- Conclusiones

C O N T E N I D O

- I INTRODUCCION A LAS MACROS DE 1-2-3**
- II MACROS CON SECUENCIAS DE TECLAS**
- III COMANDOS DEL LENGUAJE DE PROGRAMACION**
- IV MANEJO DE DATOS DENTRO DE CELDAS Y TABLAS**
- V PROCESAMIENTO INTERACTIVO DE INFORMACION**
- VI RECALCULACION Y REDESPLIEGUE**
- VII MODIFICACION DEL CONTROL DEL FLUJO DEL PROGRAMA**
- VIII MANEJO DE SUBROUTINAS**
- IX MANEJO DE MENUS**
- X CONTROLANDO ERRORES Y PROTEGIENDO APLICACIONES**
- XI PROCESANDO TEXTO Y ARCHIVOS SECUENCIALES**

I INTRODUCCION A LAS MACROS DE 1-2-3

1-2-3 CONTIENE LA CAPACIDAD DE PROGRAMACIÓN PERMITIENDO CON ELLO AUTOMATIZAR VIRTUALMENTE CUALQUIER PROCEDIMIENTO QUE PUEDA SER HECHO MANUALMENTE, DESDE LAS OPERACIONES MÁS SIMPLES HASTA LAS MÁS SOFISTICADAS. ESTA AUTOMATIZACIÓN DE PROCESOS ES LLAMADO PROGRAMACIÓN CON COMANDOS DE 1-2-3, O COMÚNMENTE LLAMADO MACROS.

PUEDE EXISTIR EL AHORRO DE DIGITACIÓN DE TECLAS O UNIFICAR UNA SECUENCIA DE PASOS SENCILLOS QUE SE EJECUTAN TODOS AL MISMO TIEMPO POR MEDIO DE UNA MACRO.

POR EJEMPLO, UNA MACRO PUEDE AUTOMÁTICAMENTE ESCRIBIR LA FECHA, EL NOMBRE DE UNA COMPANÍA O CUALQUIER OTRO TEXTO EN UN REPORTE. SE PUEDEN GENERAR CONSOLIDADOS DE INFORMACIÓN DE DIFERENTES HOJAS DE CÁLCULO O FUENTES DE INFORMACIÓN, CON DIFERENTES FORMATOS DE IMPRESIÓN PARA LA MISMA HOJA Y SIN NECESIDAD DE REPETIRLOS, ENTRE OTROS PROCESOS QUE SON SUCEPTIBLES DE SER AUTOMATIZADOS.

TAMBIÉN SE PUEDE CREAR UNA MACRO INTERACTIVA, DE TAL MODO QUE UNA VEZ QUE INICIE TENGA LA FACILIDAD DE SOLICITAR DATOS DEL USUARIO Y PUEDA CONTINUAR CON EL PROCESO. UNA MACRO INTERACTIVA EN 1-2-3 PUEDE MANEJAR UNA SESIÓN COMPLETA DE TRABAJO, DE TAL MODO QUE EL USUARIO SOLO SELECCIONE OPCIONES Y/O INTRODUZCA DATOS PARA REALIZAR SUS PROCESOS.

SE PUEDE CREAR UNA MACRO CON SOLO DIGITAR EL PROCEDIMIENTO Y ASIGNÁNDOLE UN NOMBRE DE RANGO. CUANDO SE INVOQUE LA MACRO POR MEDIO DE LA ASOCIACIÓN DEL NOMBRE, 1-2-3 RÁPIDAMENTE EJECUTA EL PROCEDIMIENTO, ACTUANDO SOBRE LOS DATOS QUE SE LOCALIZAN EN ESE MOMENTO SOBRE LA HOJA DE CÁLCULO.

DENTRO DE LA HOJA DE CÁLCULO SE PUEDEN INCLUIR HASTA 26 MACROS (26 DE LAS LETRAS DEL ALFABETO MAS LA MACRO DE AUTOINICIO). SI LA HOJA DE CÁLCULO ES SALVADA COMO ARCHIVO, ENTONCES LAS MACROS QUE ESTÁN CONTENIDAS DENTRO DE LA HOJA TAMBIÉN SON SALVADAS.

ESTRUCTURA DE UNA MACRO.-

UNA MACRO ESTÁ COMPUESTA DE UNA SECUENCIA DE TECLAS (POR EJEMPLO, A, B, /), DE TECLAS ESPECIALES (POR EJEMPLO, {EDIT}, {GRAPH}, ~) Y DE COMANDOS DEL LENGUAJE DE PROGRAMACIÓN DE 1-2-3 (POR EJEMPLO, {LET}, {IF}, {MENCALL}). ESTAS SE INTRODUCEN DIRECTAMENTE EN LAS CELDAS DE LA HOJA DE CÁLCULO COMO CUALQUIER OTRO DATO.

CADA CELDA PUEDE CONTENER UNA O MÁS MACROINSTRUCCIONES. UNA MACROINSTRUCCIÓN PUEDE SER LA DIGITACIÓN NORMAL QUE SE REALIZA DESDE EL TECLADO (COMO SE MUESTRA EN LA PRIMERA CELDA), UNA TECLA ESPECIAL (COMO EN LA CELDA 2), ENTRAR AL MENÚ Y SELECCIONAR UNA OPCIÓN (COMO EN LA CELDA 3) O UN COMANDO DEL LENGUAJE DE PROGRAMACIÓN (COMO EN LA CELDA 4).

```

+-----+
| REPORTE FINAL~ |
+-----+
| {DOWN}         |
+-----+
| /WCS25~       |
+-----+
| {BLANK TITULO}|
+-----+
| CUARTO TRIMESTRE~|
+-----+

```

UNA MACRO PUEDE TENER TECLAS, TECLAS ESPECIALES O COMANDOS

A CONTINUACIÓN SE MUESTRAN CUATRO MACROS, LAS TRES PRIMERAS SON SECUENCIAS DE TECLAS. LA CUARTA UTILIZA UNA INSTRUCCIÓN DEL LENGUAJE DE PROGRAMACIÓN.

MACRO	ACCION
'LOGICA PROGRAMADA	DIGITA LOGICA PROGRAMADA
'{GOTO}A5~75%	DIGITA 75% EN LA CELDA A5
'/fSPRES86~	SALVA LA HOJA ACTUAL BAJO EL NOMBRE DE PRES86
'{LET TOTAL,COSTO*1.15}	COLOCA EL RESULTADO DE LA MULTIPLICACIÓN DEL CONTENIDO DE LA CELDA COSTO POR 1.15 DENTRO DE LA CELDA TOTAL.

CADA UNA DE LAS CUATRO MACROS OCUPA UNA SOLA CELDA, LA PRIMERA CONSISTE DE 17 INSTRUCCIONES -SEIS LETRAS, UN ESPACIO Y OTRAS DIEZ LETRAS-. LA TERCER MACRO CONSISTE DE 10 INSTRUCCIONES -LA DIAGONAL (/) SELECCIONA EL MENÚ PRINCIPAL, FS SELECCIONA FILE SAVE Y LA TILDE (~) REPRESENTA EL RETURN-. LA CUARTA MACRO NO REPRESENTA TECLAS, ÚNICAMENTE UN COMANDO DEL LENGUAJE DE PROGRAMACIÓN.

NÓTESE QUE CADA UNA DE LAS CUATRO MACROS SON ETIQUETAS, INDICADAS POR EL PREFIJO DE ETIQUETA (AL IGUAL QUE TODAS LAS ETIQUETAS, EL PREFIJO DE ETIQUETA NO APARECE EN LA CELDA, SOLAMENTE EN EL PANEL DE CONTROL).

CUALQUIER MACRO DEBERÁ SER UN VALOR DEL TIPO STRING -LAS CELDAS NUMÉRICAS NO SON VÁLIDAS DENTRO DE UNA MACRO-. GENERALMENTE LAS MACROS SON INTRODUCIDAS COMO ETIQUETAS, SIN EMBARGO TAMBIÉN PUEDEN SER FÓRMULAS DEL TIPO STRING. AL MOMENTO DE INTRODUCIR UNA MACRO DENTRO DE UNA CELDA, QUE EMPIECE COMO ETIQUETA, 1-2-3 AUTOMÁTICAMENTE LE ADICIONA EL PREFIJO DE ETIQUETA. SI SE TRATA DE UN VALOR NUMÉRICO, COMO EN EL CASO DE LA SEGUNDA MACRO, MANUALMENTE SE TENDRÁ QUE DIGITAR EL PREFIJO DE ETIQUETA.

LA SEGUNDA MACRO EMPIEZA CON UNA TECLA QUE REPRESENTA DIGITAR LA TECLA F5 (GOTO). EN UNA MACRO TODAS LAS TECLAS ESPECIALES (INCLUYENDO LAS TECLAS DE EDICIÓN, FLECHAS Y TECLAS ASIGNADAS A FUNCIONES ESPECIFICAS DE 1-2-3) SON REPRESENTADAS POR SU NOMBRE COMPLETO ENCERRADAS ENTRE PARÉNTESIS DE LLAVES.

LA CUARTA MACRO ES UN COMANDO DEL LENGUAJE DE PROGRAMACIÓN DE 1-2-3. UN COMANDO DEL LENGUAJE DE PROGRAMACIÓN ES UNA MACROINSTRUCCIÓN QUE ESTÁ DEFINIDA DENTRO DE 1-2-3. ALGUNOS DE ESTOS COMANDOS CONSISTEN SOLO DE UNA PALABRA CLAVE, PERO OTROS TAMBIÉN CONTIENEN ARGUMENTOS QUE ACTÚAN SOBRE LA MACRO. TODOS LOS COMANDOS TAMBIÉN DEBERÁN IR ENCERRADOS ENTRE PARÉNTESIS DE LLAVE.

LAS SECUENCIAS DE TECLAS Y LOS COMANDOS SON LA PARTE MEDULAR DE ESTE MANUAL, POR LO QUE SERÁN DISCUTIDOS EN GRAN DETALLE POSTERIORMENTE.

ADEMÁS DE LAS ETIQUETAS, LAS MACROS CONTENIDAS EN LAS CELDAS PUEDEN SER TAMBIÉN FÓRMULAS DEL TIPO STRING.

LA FÓRMULA +"PERIODO "&@Sstring(NUMPER,0) ES UNA MACRO VÁLIDA, DADO QUE EL RESULTADO DE LA FÓRMULA ES UN STRING. CUANDO 1-2-3 LEE UNA CELDA QUE CONTIENE UNA MACRO, UTILIZA CADA UNO DE LOS CARACTERES COMO INSTRUCCIONES. CUANDO LA CELDA CONTIENE UNA FÓRMULA DEL TIPO STRING, 1-2-3 EVALÚA LA FÓRMULA UTILIZANDO EL RESULTADO, NO ASÍ LOS CARACTERES DE LA FÓRMULA.

'PERIODO 4 --> |-----|
PERIODO 4

MACRO INTRODUCIDA COMO ETIQUETA

+"PERIODO "&@Sstring(NUMPER,0) --> |-----|
PERIODO 4

MACRO INTRODUCIDA COMO FÓRMULA

TAMANO DE UNA MACRO.-

EL TAMANO DE UNA MACRO ESTÁ LIMITADA A LA CANTIDAD DE MEMORIA RAM. DENTRO DE UNA CELDA SE PUEDEN TENER HASTA 240 CARACTERES, LOS CUALES PUEDEN SER UNA MACRO, PERO A SU VEZ LA MACRO PUEDE ESTAR CONSTITUIDA DE VARIAS CELDAS ADYACENTES EN FORMA DE COLUMNA PARA SER PROCESADAS DE MANERA SECUENCIAL. LA COLUMNA PUEDE EMPEZAR EN CUALQUIER PARTE DE LA HOJA. CUANDO 1-2-3 TERMINA LA EJECUCIÓN DE LAS INSTRUCCIONES DE UNA CELDA, AUTOMÁTICAMENTE CONTINÚA CON LAS INSTRUCCIONES DE LA CELDA DE ABAJO. EN ESTE SENTIDO LA ESTRUCTURA DE LAS MACROS ES SEMEJANTE A LOS PROGRAMAS ESCRITOS EN OTROS LENGUAJES.

TERMINACIÓN DE UNA MACRO.-

1-2-3 LEE LAS MACROINSTRUCCIONES DENTRO DE UNA COLUMNA, CELDA POR CELDA, HACIA ABAJO HASTA QUE ENCUENTRA UNA INSTRUCCIÓN QUE FINALIZA LA MACRO O CUANDO UNA CELDA NO ES RECONOCIDA COMO UNA MACROINSTRUCCIÓN. PARA FINALIZAR UNA MACRO SE PUEDEN DAR CUALQUIERA DE LAS SIGUIENTES CONDICIONES:

- A) EL COMANDO {QUIT} DEL LENGUAJE DE PROGRAMACIÓN,
- B) UNA CELDA NUMÉRICA,
- C) UNA CELDA VACÍA O,
- D) UNA CELDA QUE APARECE VACÍA PERO QUE CONTIENE UNA FÓRMULA CUYO VALOR ES UN STRING VACÍO.

PARA SIMPLIFICAR LA TERMINACIÓN DE UNA MACRO, DENTRO DE ESTE MANUAL SE UTILIZARÁ LA CELDA VACÍA.

NOMBRANDO UNA MACRO.-

DADO QUE UNA MACRO ESTÁ CONTENIDA DENTRO DE LAS CELDAS Y PARA QUE 1-2-3 PUEDA CONOCER LA DIFERENCIA ENTRE UNA MACRO Y OTRO TIPO DE INFORMACIÓN, ES NECESARIO ASIGNARLE UN NOMBRE A LA MACRO. ÉSTO SE REALIZA ASIGNÁNDOLE UN NOMBRE DE RANGO DE CELDA A LA PRIMERA CELDA QUE CONTIENE LA MACRO, NO ES NECESARIO ASIGNARLE EL NOMBRE A TODA LA COLUMNA.

A	
2	REPORTE FINAL~
3	{DOWN}
4	/WCS25~
5	{BLANK TITULO}
6	CUARTO TRIMESTRE~

<-- PARA IDENTIFICAR ESTA COLUMNA DE CELDAS COMO UNA MACRO, SOLO SE ASIGNA EL NOMBRE DE RANGO A LA PRIMERA CELDA (A2).

EL NOMBRE DE UNA MACRO CONSISTIRÁ SOLO DE DOS CARACTERES, EL PRIMERO DEBERÁ SER LA DIAGONAL INVERTIDA (\) Y EL SEGUNDO CUALQUIER LETRA DEL ALFABETO. POR EJEMPLO, \A, \B, \Z SON NOMBRES DE MACROS VÁLIDOS.

LA ASIGNACIÓN DE LOS NOMBRES DE LAS MACROS SE PODRÁ REALIZAR CON /RANGE NAME CREATE O CON /RANGE NAME LABELS.

ADICIONALMENTE SE TIENE UNA MACRO CUYO NOMBRE ES ESPECIAL, LA MACRO \0, LA CUAL SE AUTOEJECUTA AUTOMÁTICAMENTE CADA VEZ QUE SE CARGA LA HOJA DE CÁLCULO CON /FILE RETRIEVE.

DOCUMENTANDO UNA MACRO.-

ES MUY ÚTIL DOCUMENTAR UNA MACRO, UTILIZANDO SIMPLEMENTE UNA ETIQUETA CON EL NOMBRE DE LA MACRO A LA IZQUIERDA DEL INICIO DE ÉSTA. ESTA DOCUMENTACIÓN PROPORCIONA CLARIDAD, NO ES NECESARIO TENER EN MENTE LOS NOMBRES DE CADA UNA DE LAS MACROS. ADEMÁS AL METER EL NOMBRE DE LA MACRO A LA IZQUIERDA, SE PUEDE UTILIZAR LA INSTRUCCIÓN RANGE NAME LABELS RIGHT PARA ASIGNAR EL NOMBRE A LA MACRO.

	A	B	
NOMBRE DE LA MACRO -->	\A	REPORTE FINAL~	<-- PRIMERA CELDA DE LA MACRO
3		{DOWN}	
4		/WCS25~	
5		{BLANK TITULO}	
6		CUARTO TRIMESTRE~	

SE PUEDE UTILIZAR /RANGE NAME LABELS RIGHT PARA ASOCIAR EL NOMBRE DE LA MACRO (\A) A LA PRIMERA CELDA DE LA MACRO (B2). DE ESTA FORMA TAMBIÉN SE CREA UNA BREVE DOCUMENTACIÓN.

INVOCANDO UNA MACRO.-

UNA MACRO PUEDE SER VIRTUALMENTE EJECUTADA EN CUALQUIER MOMENTO DURANTE UNA SESIÓN DE 1-2-3. EL NOMBRE DE LA MACRO ES POR MEDIO DEL CUAL SE INVOCARÁ.

UNA MACRO BAUTIZADA (UNA DIAGONAL INVERTIDA SEGUIDA DE UNA LETRA, POR EJEMPLO, \A) ES INVOCADA CON LA COMBINACIÓN DE LA TECLA ALT Y LA LETRA ASOCIADA AL NOMBRE (POR EJEMPLO ALT-A).

ADICIONALMENTE A LA INVOCACIÓN MANUAL DE UNA MACRO, TAMBIÉN SE PUEDE TENER UNA MACRO QUE SE EJECUTE CADA VEZ QUE LA HOJA DE CÁLCULO SEA RECUPERADA CON FILE RETRIEVE, A ESTA MACRO SE LE BAUTIZARÁ COMO \0. O TAMBIÉN SE PUEDE EJECUTAR UNA MACRO QUE SEA LLAMADA POR OTRA, SIENDO ESTO ANÁLOGO A LOS PROGRAMAS QUE SON EJECUTADOS POR OTROS, COMO EN EL CASO DE C, PASCAL, ETC.

UNA MACRO NO PODRÁ SER INVOCADA CUANDO:

- + SE TIENE UNA CONDICIÓN DE ERROR (EL INDICADOR DE MODO ESTÁ EN ERROR),
- + SE TIENE HELP O,
- + SE EJECUTA OTRA MACRO.

INTERRUMPIENDO LA EJECUCIÓN DE UNA MACRO.-

DESDE EL MOMENTO QUE UNA MACRO INICIA HASTA QUE TERMINA, EL CONTROL DE LA SESIÓN DE 1-2-3 PASA A UN PROCESO AUTOMATIZADO. UNA TÍPICA MACRO SE EJECUTA DURANTE VARIOS SEGUNDOS Y AL FINALIZAR REGRESA EL CONTROL AL USUARIO. SIN EMBARGO, EN OCASIONES EL USUARIO NECESITA TOMAR EL CONTROL ANTES DE QUE LA MACRO FINALICE, O CUANDO SE TIENE UN LOOP.

AL PRESIONAR CTRL-BREAK EL PROCESO DE LA MACRO FINALIZA, REGRESANDO EL CONTROL AL USUARIO, PARA QUE ÉSTE DIGITE LO NECESARIO DESDE EL TECLADO. EN ALGUNAS OCASIONES NO ES NECESARIO SOLAMENTE PRESIONAR EL CTRL-BREAK, SINO QUE SERÁ NECESARIO REALIZARLO DOS VECES, POR EJEMPLO, CUANDO LA MACRO ESTÁ EN PROCESO DE IMPRESIÓN, EL PRIMER CTRL-BREAK, SERVIRÁ PARA SUSPENDER LA IMPRESIÓN Y EL SEGUNDO PARA SUSPENDER LA MACRO.

EL LENGUAJE DE COMANDOS TIENE LA FACILIDAD PARA HABILITAR O DESHABILITAR LA OPCIÓN DE CTRL-BREAK. PERMITIENDO CON ESTO CONTROLAR A OTROS USUARIOS EL ACCESAR UNA MACRO.

REVISANDO LA EJECUCIÓN DE UNA MACRO PASO A PASO.

EL MODO DE PASO A PASO ES ÚTIL PARA REALIZAR UNA PRUEBA DE LA MACRO, DE ESTA MANERA EL PROGRAMADOR ESTARÁ SEGURO QUE EL PROGRAMA FUNCIONARÁ CORRECTAMENTE.

EL MODO DE PASO A PASO SE ACTIVA OPRIMIENDO AL MISMO TIEMPO LA TECLA SHIFT Y F2, DESPUÉS APARECERÁ EN LA PARTE INFERIOR DE LA PANTALLA EL INDICADOR SST. UNA VEZ QUE ESTE ACTIVADO ESTE MODO SE PUEDE EJECUTAR LA MACRO Y SE TENDRÁ QUE OPRIMIR ALGUNA TECLA PARA QUE LA EJECUCIÓN DE LA MACRO CONTINÚE. PARA MAYOR FACILIDAD SE RECOMIENDA USAR LAS TECLAS RETURN O BARRA ESPACIADORA.

ESTE MODO SE PUEDE DESACTIVAR OPRIMIENDO NUEVAMENTE LAS TECLAS SHIFT Y F2

USANDO CELDAS COMO VARIABLES DE DATOS.

CUALQUIER LENGUAJE DE PROGRAMACIÓN TRABAJA CON VARIABLES DE DATOS. UNA VARIABLE ES UNA LOCALIDAD DONDE SE PUEDE ALMACENAR UN DATO Y A LA CUAL SE LE ASIGNA UN NOMBRE. EN 1-2-3 UNA CELDA ES UNA VARIABLE. CADA CELDA PUEDE ALMACENAR YA SEA UN VALOR NUMÉRICO O UN STRING. LAS VARIABLES PUEDEN ESTAR LOCALIZADAS EN CUALQUIER PARTE DE LA HOJA DE CÁLCULO, DE MANERA DIFERENTE A ALGUNOS LENGUAJES DE PROGRAMACIÓN. SIN EMBARGO, ES RECOMENDABLE TENER UN ÁREA DONDE ESTÉN LOCALIZADAS.

DIRECCIONES DE CELDA CONTRA NOMBRES DE RANGO.-

MUCHOS LENGUAJES DE PROGRAMACIÓN DISTINGUEN LAS VARIABLES DE DATOS ASIGNÁNDOLES DIFERENTES NOMBRES. EN LAS MACROS DE 1-2-3, LAS VARIABLES DE DATOS SON CELDAS DENTRO DE LA HOJA DE CÁLCULO QUE PUEDEN SER IDENTIFICADAS CON LA DIRECCIÓN DE LA CELDA. ESTO ES PERMITIDO, SIN EMBARGO, ES PREFERIBLE ASIGNAR UN NOMBRE DE RANGO A CADA VARIABLE. AL REORGANIZAR LA HOJA DE CÁLCULO, GENERALMENTE LAS MACROS NO SON INVALIDADAS CUANDO SE UTILIZAN NOMBRES DE RANGO, PERO AL UTILIZAR LAS DIRECCIONES DE CELDA, GENERALMENTE LAS MACROS SON AFECTADAS.

{LET TOTAL,10.2}	RECOMENDADO
{LET A10,10.2}	NO RECOMENDADO

AL UTILIZAR UN NOMBRE DE RANGO SE DEBE TOMAR EN CUENTA, QUE NO ES PARTE DE LA MACRO, SIMPLEMENTE ES PARTE DE LA HOJA DE CÁLCULO Y SERVIRÁ PARA EVITAR ERRORES EN LAS MODIFICACIONES Y PARA MAYOR CLARIDAD. ES CONVENIENTE QUE LOS NOMBRES SE ASEMEJEN A LA INFORMACIÓN QUE CONTENDRÁN.

II MACROS CON SECUENCIAS DE TECLAS

AL DIGITAR UNA SECUENCIA DE TECLAS DE MANERA FRECUENTE, ES NECESARIO QUE NO SE VUELVA A REPETIR LA DIGITACIÓN. PARA ELLO DICHA SECUENCIA SERÁ ALMACENADA DENTRO DE LA HOJA DE CÁLCULO COMO UNA MACRO. CUANDO LA MACRO SEA INVOCADA, 1-2-3 LEE LA SECUENCIA DE TECLAS, UNA POR UNA, Y PROCESA LA SECUENCIA, DE TAL MODO QUE SE ESTUVIERA DIGITANDO DESDE EL TECLADO NUEVAMENTE.

```

+-----+
| TOTAL~ |
+-----+
| {RIGHT}|
+-----+
| /WCS12~|
+-----+

```

SECUENCIA DE TECLAS INDICADAS POR LA MACRO:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|T|O|T|A|L| |R|E|T|U|N| |->| |/|W|C|S|1|2| |R|E|T|U|N|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
          ^           ^
          |           |

```

INDICADAS POR ~ {RIGHT}

UNA MACRO PROCESA UNA SECUENCIA DE TECLAS AUTOMÁTICAMENTE

ESTA CAPACIDAD DE "DIGITACIÓN AUTOMÁTICA" PUEDE SER APLICADA A CUALQUIER OPERACIÓN QUE SE REALICE EN 1-2-3. USOS TÍPICOS DE LAS SECUENCIAS DE TECLAS SON: INTRODUCIR ETIQUETAS, NÚMEROS O FÓRMULAS DENTRO DE LA HOJA, IMPRESIÓN DE UN REPORTE, SALVAR LA HOJA DE CÁLCULO, ETC.

LAS TECLAS DENTRO DE UNA MACRO DEL TIPO SECUENCIA DE TECLAS SON REPRESENTADAS EN DOS FORMAS. LA PRIMERA ES UNA TECLA QUE DESPLIEGA UN CARACTER EN LA PANTALLA O SELECCIONA UN ELEMENTO DE UN MENÚ, SIENDO REPRESENTADA POR UN SOLO CARACTER. LA SEGUNDA SON LAS TECLAS ESPECIALES QUE SON REPRESENTADAS POR SU NOMBRE ENCERRADO DENTRO DE PARÉNTESIS DE LLAVE (POR EJEMPLO, {LEFT}, {GOTO}, {WINDOW}, ENTRE OTROS.).

PREFIJOS DE ETIQUETA EN UNA MACRO.-

EN UNA MACRO, 1-2-3 UTILIZA CADA CELDA COMO UN VALOR TIPO STRING, MAS QUE EL CONTENIDO DE LA CELDA EN SÍ. EL VALOR TIPO STRING DE UNA CELDA INCLUYE TODOS LOS CARACTERES INTRODUCIDOS A EXCEPCIÓN

DEL PREFIJO DE ETIQUETA. ESTO ES, CUANDO SE EJECUTA UNA SECUENCIA DE TECLAS ALMACENADAS EN UNA COLUMNA DE CELDAS, 1-2-3 IGNORA CADA PREFIJO DE ETIQUETA DE LA CELDA. POR EJEMPLO, ES LO MISMO UNA MACRO QUE CONTenga EL PREFIJO DE CENTRADO (^), QUE EL PREFIJO DE JUSTIFICACIÓN A LA IZQUIERDA.

```

+-----+
|         |
| REPORTE |
|         |
| TRIMESTRAL |
|         |
+-----+

```

ETIQUETAS A LA IZQUIERDA

```

+-----+
|         |
|   REPORTE   |
|         |
|   TRIMESTRAL   |
|         |
+-----+

```

ETIQUETAS CENTRADAS

EL PREFIJO DE ETIQUETA NO ES CONSIDERADO EN LA EJECUCIÓN DE LA MACRO

CONTINUACIÓN DE UNA MACRO EN UNA COLUMNA.-

1-2-3 LEE CADA SECUENCIA DE TECLAS COMO UNA MACRO INDEPENDIENTE. ESTO SIGNIFICA QUE ES LO MISMO METER LARGAS SECUENCIAS DE MACROS DENTRO DE UNA CELDA (HASTA 240 CARACTERES) QUE PARTIR LAS SECUENCIAS EN VARIAS CELDAS CONSECUTIVAS EN FORMA DE COLUMNA. DE HECHO, ES RECOMENDABLE TENER SECUENCIAS CORTAS PARA UNA MEJOR LEGIBILIDAD DE LAS MACROS Y DETECCIÓN DE ERRORES.

CUANDO 1-2-3 TERMINA DE LEER LA MACRO QUE SE LOCALIZA EN UNA CELDA, INMEDIATAMENTE CONTINUA CON LA LECTURA DE LA SIGUIENTE CELDA. LAS TRES MACROS QUE SE MUESTRAN A CONTINUACIÓN SON EXACTAMENTE IGUALES.

```

+-----+ +-----+ +-----+
| LOGICA PROGRAMADA | | LOGICA | | LOGI |
+-----+ +-----+ +-----+
|         | | PROGRAMADA | | CA PROGRA |
+-----+ +-----+ +-----+
|         | |         | | MADA |
+-----+ +-----+ +-----+

```

LAS INSTRUCCIONES EN 1-2-3 SE LEEN SECUENCIALMENTE

LAS TECLAS ESPECIALES Y LOS COMANDOS DEL LENGUAJE DE PROGRAMACIÓN NO PUEDEN SER DIVIDIDOS. 1-2-3 LEE CUALQUIER CONJUNTO DE CARACTERES ENCERRADOS ENTRE LLAVES ({}), COMO UNA SOLA MACROINSTRUCCIÓN. POR CONSIGUIENTE, ESTE TIPO DE TECLAS ESPECIALES NO PUEDEN SER DIVIDIDAS EN DOS O MÁS CELDAS. EL

SIGUIENTE EJEMPLO MANDARÍA UN ERROR, DADO QUE LA ÚLTIMA MACROINSTRUCCIÓN ({RIGHT}) ESTÁ DIVIDIDA EN DOS CELDAS.

```
-----+-----  
|/C~{UP}{RIGHT}{RI|  
-----+-----  
|GHT}~|  
-----+-----
```

LAS MACROINSTRUCCIONES DIVIDIDAS EN DOS O MÁS CELDAS GENERAN ERROR

LA TECLA RETURN.-

LA TECLA RETURN O ENTER ES PROBABLEMENTE LA TECLA MÁS UTILIZADA DENTRO DE UNA SESIÓN EN 1-2-3. DE LA MISMA MANERA QUE EN LAS MACROS. RETURN ES LA ÚNICA TECLA ESPECIAL QUE NO ES REPRESENTADA EN UNA MACRO CON SU NOMBRE ENCERRADO ENTRE PARÉNTESIS DE LLAVE.

SU REPRESENTACIÓN DENTRO DE UNA MACRO SE HACE POR MEDIO DE LA TILDE (~). EL USO DE UN SOLO CARACTER PARA REPRESENTAR ES CONVENIENTE, DADO QUE COMO SE MENCIONÓ ES UNA TECLA MUY UTILIZADA. SE DEBE TENER MUCHO CUIDADO AL CREAR LAS MACROS DE NO OLVIDAR LA TILDE CUANDO SEA REQUERIDA. SE DEBE TENER EN CUENTA QUE TAMBIÉN AL CAMBIAR DE UNA CELDA A OTRA NO SE ESTA INDICANDO UN RETURN. POR EJEMPLO, LA SIGUIENTE MACRO NO SE EJECUTARÍA CORRECTAMENTE, DADO QUE NO SE INDICA EL RETURN.

```
'LOTUS DEVELOPMENT CORPORATION  
'/FSPRODUCTO
```

ESTA MACRO AL INVOCARSE GENERARÍA LA ETIQUETA LOTUS DEVELOPMENT CORPORATION/FSPRODUCTO. LA MACRO CORRECTA PARA REALIZAR LA GRABACIÓN DEL ARCHIVO PRODUCTO SERÍA:

```
'LOTUS DEVELOPMENT CORPORATION~  
'/FSPRODUCTO~
```

AL SER REPRESENTADA LA TILDE COMO LA TECLA RETURN SE PUEDE PRESENTAR UNA RESTRICCIÓN AL QUERER INCLUIR LA TILDE COMO CARACTER DENTRO DE UNA MACRO, PARA SOLUCIONAR ESTE PROBLEMA, 1-2-3 REPRESENTA LA TILDE CON LA SECUENCIA {~}.

TECLAS ESPECIALES.-

LAS DEMÁS TECLAS ESPECIALES NO PUEDEN SER REPRESENTADAS ÚNICAMENTE CON UN SOLO CARACTER. EN LUGAR DE ESO, SE REPRESENTARÁN CON EL NOMBRE DE LA TECLA (O EN ALGUNOS CASOS CON ABREVIACIONES) ENCERRADO ENTRE PARÉNTESIS DE LLAVE. SE PUEDE UTILIZAR LETRA MAYÚSCULA O MINÚSCULA:

'PRIMERO~{RIGHT}SEGUNDO{RIGHT}TERCERO{HOME}

LA MACRO ANTERIOR INCLUYE UN RETURN, DOS DESPLAZAMIENTOS A LA DERECHA Y LA TECLA HOME.

LA SIGUIENTE TABLA MUESTRA TODAS LAS TECLAS ESPECIALES Y SUS EQUIVALENTES QUE PUEDEN SER UTILIZADAS DENTRO DE UNA MACRO DE 1-2-3.

BACKSPACE	{BACKSPACE}	(BORRA CARAC. IZQUIERDO)
CTRL-LEFT	{BIGLEFT}	(PANTALLA IZQUIERDA)
CTRL-RIGHT	{BIGRIGHT}	(PANTALLA DERECHA)
CALC	{CALC}	(TECLA F9)
DOWN	{DOWN}	(CELDA DE ABAJO)
EDIT	{EDIT}	(TECLA F2)
END	{END}	(FIN DE RENG. O COL.)
ESCAPE	{ESCAPE}	(ESCAPE)
GOTO	{GOTO}	(TECLA F5)
HOME	{HOME}	(INICIO DE LA HOJA)
LEFT	{LEFT}	(CELDA IZQUIERDA)
PAGEDOWN	{PGDN}	(PANTALLA DE ABAJO)
PAGEUP	{PGUP}	(PANTALLA DE ARRIBA)
RETURN	~	(TRANSMITE)
RIGHT	{RIGHT}	(CELDA DERECHA)
UP	{UP}	(CELDA DE ARRIBA)

TECLAS ESPECIALES QUE NO PUEDEN SER INTRODUCIDAS EN LAS MACROS.-

LAS SIGUIENTES TECLAS ESPECIALES NO PUEDEN SER INCLUIDAS DENTRO DE UNA MACRO:

CAPS LOCK. ACTIVA LAS LETRAS COMO MAYÚSCULAS O COMO MINÚSCULAS. DENTRO DE UNA MACRO NO SE PODRÁ REALIZAR ESTE CAMBIO. LOS CARACTERES QUE SE DESEEN SERÁN DIGITADOS DE LA MANERA DESEADA:

'{CAPS LOCK}ABCD <-- MACRO INCORRECTA

'ABCD <-- MACRO CORRECTA

NUM LOCK. ACTIVA LA FUNCIÓN DE TECLADO NUMÉRICO O DE DESPLAZAMIENTO. COMO EN EL CASO DE CAPS LOCK LA MACRO DETERMINARÁ EL CAMBIO DE ESTADO DEL TECLADO.

SCROLL LOCK. DETERMINA EL TIPO DE DESPLAZAMIENTO DENTRO DE LA PANTALLA. COMO EN EL CASO DE CAPS LOCK LA MACRO DETERMINARÁ EL ESTADO DE SCROLL LOCK.

STEP. CONTROLA LA EJECUCIÓN DE UNA MACRO PARA PROCESARSE EN PAUSAS INSTRUCCIÓN POR INSTRUCCIÓN. ÉSTA FACILIDAD ES ÚTIL PARA REALIZAR EL DEBUG DE UNA MACRO, SIN EMBARGO NO PUEDE SER ACTIVADA DESDE UNA MACRO.

COMPOSE. DA LA FACILIDAD PARA QUE CUALQUIER CARACTER DE LICS SEA DIGITADO DESDE EL TECLADO (POR EJEMPLO, PARA LETRAS CON ACENTO). DENTRO DE LAS MACROS NO SE PUEDE INCLUIR:

'AN{COMPOSE}ALISIS TOTAL~ <-- MACRO INCORRECTA
'ANÁLISIS TOTAL~ <-- MACRO CORRECTA

FACTORES DE REPETICIÓN.-

UNA MACRO PUEDE INCLUIR LA DIGITACIÓN CONSECUTIVA DE LA MISMA TECLA. POR EJEMPLO, SE PUEDE DESEAR MOVER CUATRO CELDAS A LA DERECHA PARA DIGITAR UN NUEVO DATO DENTRO DE UNA CELDA. EN LUGAR DE DIGITAR {RIGHT} CUATRO VECES, SE PUEDE UTILIZAR EL FACTOR DE REPETICIÓN: {RIGHT 4}.

LAS SIGUIENTES REGLAS SE APLICAN AL FACTOR DE REPETICIÓN:

- A) EL NOMBRE DE LA MACRO DEBERÁ SER SEGUIDO EXACTAMENTE POR UN ESPACIO. SI SE DIGITA CUALQUIER OTRO NÚMERO DE ESPACIOS LA MACRO TERMINARÁ MANDANDO EL MENSAJE DE ERROR: "UNRECOGNIZED KEY/RANGE NAME" O "MACRO: ILLEGAL KEY REPEAT COUNT"
- B) EL FACTOR DE REPETICIÓN PODRÁ SER UN NÚMERO, UN NOMBRE DE RANGO O UNA EXPRESIÓN NUMÉRICA. SI SE UTILIZA UN NOMBRE DE RANGO O UNA EXPRESIÓN NUMÉRICA, 1-2-3 UTILIZA EL VALOR AL MOMENTO DE QUE LA MACRO ES EJECUTADA, NO SIENDO ESTE VALOR NECESARIAMENTE IGUAL AL MOMENTO DE CREAR LA MACRO.

MACRO	ACCION
'{DOWN 6}	PRESIONA LA TECLA DOWN SEIS VECES
'{UP SUBE+2}	PRESIONA LA TECLA UP TANTAS VECES COMO SE INDICA EN LA EXPRESIÓN SUBE+2

III COMANDOS DEL LENGUAJE DE PROGRAMACION

MUCHAS DE LA TAREAS DE PROCESAMIENTO DE DATOS PUEDEN SER REALIZADAS EJECUTANDO SIMPLES SECUENCIAS DE TECLAS -TRANSFERIR DATOS DE UN RANGO A OTRO, BORRAR DATOS DE LAS CELDAS, CREAR Y EDITAR NUEVOS DATOS EN LAS CELDAS-, TAMBIÉN PUEDEN SER EJECUTADAS POR MEDIO DE LOS COMANDOS DEL LENGUAJE DE PROGRAMACIÓN DE 1-2-3. EN MUCHOS CASOS ES MAS CONVENIENTE Y RÁPIDO EJECUTAR LOS COMANDOS QUE UTILIZAR LAS SECUENCIAS DE TECLAS. PERO LOS COMANDOS NO SON ÚNICAMENTE UN MÉTODO ALTERNATIVO PARA REALIZAR LOS PROCEDIMIENTOS QUE SE REALIZAN CON LAS SECUENCIAS DE TECLAS. SINO QUE TAMBIÉN PUEDEN SER UTILIZADOS EN COMBINACIÓN CON LAS SECUENCIAS, PARA REALIZAR TAREAS MÁS SOFISTICADAS QUE NO PODRÍAN SER REALIZADAS SIN LA EXISTENCIA DE ESTOS COMANDOS. LOS COMANDOS DEL LENGUAJE DE PROGRAMACIÓN PERMITEN REALIZAR MACROS QUE INCLUYEN LAS SIGUIENTES CARACTERÍSTICAS:

- A) PROCESAMIENTO INTERACTIVO. LOS COMANDOS PERMITEN REALIZAR MACROS INTERACTIVAS -PROCESOS QUE REALIZAN PAUSAS PARA QUE EL USUARIO INTRODUZCA INFORMACIÓN DESDE EL TECLADO, CONTINUANDO CON EL PROCESO-. UNA MACRO INTERACTIVA PUEDE REALIZAR UNA PAUSA PARA INTRODUCIR UNA SIMPLE TECLA, UN DATO O POR UN PERÍODO INDEFINIDO. TAMBIÉN PUEDE INVOCAR MENÚS DENTRO DEL PANEL DE CONTROL Y CONTINUAR EL PROCESO EN BASE A LA OPCIÓN SELECCIONADA.
- B) CONTROL EN EL DESPLIEGUE DE LA PANTALLA. ADEMÁS DE PODER CONTROLAR EL TECLADO, UNA MACRO PUEDE CONTROLAR EL DESPLEGADO DE LA PANTALLA, YA SEA CAMBIANDO EL DESPLIEGUE DEL INDICADOR DE MODO, CONGELANDO LA PANTALLA DURANTE EL PROCESAMIENTO DE UNA MACRO O SIMPLEMENTE ENVIANDO UN SONIDO PARA LLAMAR LA ATENCIÓN DEL USUARIO.
- C) CONTROLAR EL FLUJO DE INSTRUCCIONES QUE SE EJECUTAN EN LA MACRO. EL CONTROL DEL FLUJO DEL PROGRAMA DENTRO DE UNA MACRO DE 1-2-3 LO CONVIERTE EN UN COMPLETO Y PODEROSO LENGUAJE DE PROGRAMACIÓN. LOS COMANDOS UTILIZADOS PARA CONTROLAR EL FLUJO DEL PROGRAMA SON MUY SIMILARES A LOS UTILIZADOS EN LOS MAS POPULARES LENGUAJES DE PROGRAMACIÓN:

IF-THEN	INSTRUCCIÓN CONDICIONAL
GOTO	INSTRUCCIÓN PARA CAMBIAR EL FLUJO DEL PROGRAMA
FOR-NEXT	INSTRUCCIÓN PARA SECUENCIAS REPETITIVAS (LOOPS)
SUBROUTINAS	LLAMADO A MÓDULOS INDEPENDIENTES CON PASO DE PARÁMETROS

- D) ACCESO DE ARCHIVOS SECUENCIALES EXTERNOS. UNA MACRO PUEDE TRANSFERIR INFORMACIÓN ENTRE LA HOJA DE CÁLCULO Y UN ARCHIVO SECUENCIAL EN DISCO. ESTA CAPACIDAD ES TOTALMENTE INDEPENDIENTE DE LAS OPCIONES FILE Y PRINT DE 1-2-3.

TODOS LOS COMANDOS QUE SE TIENEN EN 1-2-3 TIENEN UN NOMBRE CLAVE, EL CUAL SE DERIVA DE LA ACCIÓN QUE EJECUTAN. EN ALGUNOS CASOS EL COMANDO CONSISTE ÚNICAMENTE DE SU NOMBRE CLAVE ENCERRADO ENTRE PARÉNTESIS DE LLAVE. EN MUCHOS CASOS, SIN EMBARGO, UNO O MAS ARGUMENTOS SON REQUERIDOS DENTRO DEL COMANDO. POR EJEMPLO, LA MACROINSTRUCCIÓN {LET RESULTADO,500} ES EL COMANDO {LET}, ESTÁ COMPUESTA DE SU NOMBRE CLAVE LET Y DE DOS ARGUMENTOS.

SINTAXIS DE UN COMANDO DEL LENGUAJE DE PROGRAMACIÓN.-

TODOS LOS COMANDOS DENTRO DE 1-2-3 TIENEN CUALQUIERA DE LAS DOS SIGUIENTES FORMAS:

{NOMBRE_CLAVE}

{NOMBRE_CLAVE ARGUMENTO, ARGUMENTO, ...}

DONDE: NOMBRE_CLAVE

ES UNO DE LOS COMANDOS

ARGUMENTO, ARGUMENTO, ... ES UNO O MAS ARGUMENTOS SEPARADOS POR COMA (,), PUNTO Y COMA (;) O CUALQUIER OTRO SEPARADOR VÁLIDO. EL NOMBRE_CLAVE Y EL PRIMER ARGUMENTO DEBERÁN IR SEPARADOS POR UN SOLO ESPACIO.

LAS SIGUIENTES REGLAS SE APLICAN A LOS COMANDOS:

A) CADA COMANDO DEBERÁ ESTAR DENTRO DE UNA SOLA CELDA. POR CONSIGUIENTE NO PODRÁ SER DIVIDIDO EN DOS O MAS CELDAS CONSECUTIVAS.

'{LET GANANCIA, VENTAS-GASTOS}' <-- ESTRUCTURA CORRECTA

'{LET GANANCIA,
VENTAS-GASTOS}' <-- ESTRUCTURA INCORRECTA

B) CADA COMANDO DEBERÁ EMPEZAR CON PARÉNTESIS DE LLAVE IZQUIERDO Y FINALIZAR CON EL PARÉNTESIS DE LLAVE DERECHO. ALGUNOS COMANDOS CONSISTEN SOLAMENTE DEL NOMBRE CLAVE SIN ARGUMENTOS. ESTOS COMANDOS SON DIFERENTES A LAS TECLAS ESPECIALES.

{BREAKOFF} {PANELON} {BEEP} <-- COMANDO SIN ARGUMENTOS

{CALC} {PGDN} {RIGHT} <-- TECLAS ESPECIALES

C) NO SON PERMITIDOS ESPACIOS ADICIONALES DENTRO DE UN COMANDO. SOLO SE UTILIZA EL ESPACIO PARA SEPARAR EL NOMBRE_CLAVE DE LOS ARGUMENTOS. EN CASO DE NO TENER ARGUMENTOS, NO DEBERÁN EXISTIR ESPACIOS DENTRO DEL COMANDO. NÓTESE QUE LA LISTA DE ARGUMENTOS NO SON SEPARADOS CON EL ESPACIO - LA ÚNICA POSIBILIDAD DE QUE EXISTAN ESPACIOS EN BLANCO DENTRO DE LA LISTA DE ARGUMENTOS,

ES CUANDO EL ESPACIO ES PARTE DEL MISMO ARGUMENTO (POR EJEMPLO, DENTRO DE UN STRING)-.

- D) LOS ARGUMENTOS SERÁN SEPARADOS CON UNA COMA (,); PUNTO Y COMA (;) U OTRO SEPARADOR DESIGNADO. POR SIMPLICIDAD EL SEPARADOR QUE SE UTILIZARÁ DENTRO DE ESTE MANUAL SERÁ LA COMA (,).

USANDO NOMBRES DE RANGO COMO ARGUMENTOS.-

CUANDO SE UTILIZA UN ARGUMENTO DENTRO DE UN COMANDO, A MENUDO SE USA EL NOMBRE DE RANGO DE LA CELDA QUE CONTIENE EL VALOR APROPIADO PARA EL COMANDO. POR EJEMPLO, LA SIGUIENTE MACRO USA EL VALOR DE LA CELDA COSTO PARA ALMACENAR UN NUEVO DATO DENTRO DE LA CELDA PRECIO:

```
'{LET PRECIO,COSTO*1.5}
```

ALGUNOS COMANDOS NO ACEPTAN NOMBRES DE RANGO COMO ARGUMENTOS. POR EJEMPLO, PARA CAMBIAR EL INDICADOR DE MODO A INICIA, SE DEBERÁ INCLUIR EL STRING INICIA DENTRO DEL COMANDO {INDICATE}.

```
'{INDICATE "INICIA"}
```

A TRAVÉS DE ESTE MANUAL SE UTILIZARÁN NOMBRES DE RANGOS DE CELDAS PARA MAYOR CLARIDAD EN LOS CASOS QUE SEA POSIBLE MANEJARLOS. EN LA MEDIDA DE LO POSIBLE NO SE UTILIZARÁN DIRECCIONES DE CELDA.

CUANDO SE ENCIERRA ENTRE COMILLAS UN ARGUMENTO, ESTE ES MANEJADO COMO STRING, POR EJEMPLO:

```
'{LET RESULTADO,500}           ALMACENA EL NÚMERO 500  
'{LET RESULTADO,"500"}       ALMACENA LA ETIQUETA 500  
'{LET RESULTADO,TOTAL}      ALMACENA EL VALOR DE LA CELDA TOTAL  
'{LET RESULTADO,"TOTAL"}    ALMACENA LA ETIQUETA TOTAL
```

UN ARGUMENTO DENTRO DE 1-2-3 PUEDE SER MANEJADO DIRECTAMENTE COMO STRING ANADIENDO ÚNICAMENTE EL SUFIJO :STRING AL ARGUMENTO. POR EJEMPLO:

```
'{LET COSTO,1000.50}          ALMACENA EL NÚMERO 1000.50  
'{LET COSTO,1000.50:STRING}  ALMACENA LA ETIQUETA 1000.50  
'{LET COSTO,DIR&INDIR}      ALMACENA EL RESULTADO DIR&INDIR  
'{LET COSTO,DIR&INDIR:STRING} ALMACENA LA ETIQUETA DIR&INDIR
```

IV MANEJO DE DATOS DENTRO DE CELDAS Y TABLAS

CUANDO SE UTILIZA EL LENGUAJE DE PROGRAMACIÓN EN 1-2-3 PARA EL PROCESAMIENTO DE DATOS, ES IMPORTANTE SABER COMO SE ALMACENAN LOS DATOS. CADA VEZ QUE SE REALIZA UNA SESIÓN DENTRO DE 1-2-3 SE MANIPULA LA INFORMACIÓN DE LAS CELDAS (NÚMEROS, ETIQUETAS Y FÓRMULAS). LAS MACROS PUEDEN REALIZAR LAS MISMAS ACTIVIDADES, EN ESTA SECCIÓN SE VERÁ COMO UTILIZAR LOS COMANDOS DE PROGRAMACIÓN O LAS SECUENCIAS DE TECLAS PARA ALMACENAR Y EDITAR INFORMACIÓN DE UNA CELDA. CADA MÉTODO TIENE SUS VENTAJAS, POR LO QUE EL PROGRAMADOR PODRÁ DETERMINAR CUAL DE LAS DOS TÉCNICAS ES MEJOR EN DETERMINADOS CASOS.

TAMBIÉN SE DISCUTIRÁ COMO BORRAR UNA CELDA CON EL COMANDO {BLANK} O CON LA SECUENCIA DE TECLAS /RANGE ERASE.

EL RECÁLCULO O DESPLIEGUE DE INFORMACIÓN DESPUÉS DE LA EJECUCIÓN DE UNA MACRO O CUANDO SE MODIFICA LA INFORMACIÓN EN UNA CELDA, NO NECESARIAMENTE ACTUALIZA LA INFORMACIÓN EN LA PANTALLA, PARA ELLO SE MANEJA EL {CALC} DESDE LAS SECUENCIAS DE TECLAS O EL {RECALC} O {RECALCCOL} CON LOS COMANDOS DE PROGRAMACIÓN.

INTRODUCIENDO Y EDITANDO LA INFORMACIÓN EN UNA CELDA. -

PARA PODER INTRODUCIR UN NÚMERO O UNA ETIQUETA DENTRO DE UNA CELDA, SE PUEDE UTILIZAR EL COMANDO {LET} O UNA SECUENCIA DE TECLAS. LOS SIGUIENTES EJEMPLOS MUESTRAN AMBOS MÉTODOS. LOS DOS EJEMPLOS EN LA PRIMERA LÍNEA INTRODUCEN EL VALOR 14.5% EN UN RANGO DENOMINADO TOTAL. LOS DOS EJEMPLOS EN LA SEGUNDA LÍNEA INTRODUCEN LA ETIQUETA TOTAL MENSUAL.

COMANDO {LET}	SECUENCIA DE TECLAS
{LET TOTAL, 14.5%}	{GOTO}TOTAL~14.5%~
{LET TOTAL, "TOTAL MENSUAL"}	{GOTO}TOTAL~^TOTAL MENSUAL~

EXISTE UNA DIFERENCIA IMPORTANTE AL UTILIZAR EL COMANDO {LET}, DADO QUE LAS ETIQUETAS SIEMPRE LAS ALINEARÁ A LA IZQUIERDA. UNA SECUENCIA DE TECLAS ES MAS FLEXIBLE EN ESE ASPECTO, DADO QUE SE PUEDE INCLUIR EL PREFIJO DE ETIQUETA DENTRO DE LA MACRO. NÓTESE QUE EN LA SEGUNDA LÍNEA SE COLOCA EL PREFIJO DE CENTRADO (^) EN LA SECUENCIA DE TECLAS. EN CASO DE INCLUIRSE EN EL COMANDO {LET}, ENTONCES SE TOMARÍA COMO EL PRIMER CARACTER DE LA ETIQUETA.

OTRA IMPORTANTE DIFERENCIA ENTRE EL COMANDO {LET} Y LA SECUENCIA DE TECLAS, ES QUE EL COMANDO {LET} NO PUEDE DEJAR FÓRMULAS DENTRO DE LAS CELDAS, DADO QUE ÉSTAS SON EVALUADAS Y POSTERIORMENTE EL RESULTADO ES COLOCADO EN LA CELDA CORRESPONDIENTE. CON LA SECUENCIA DE TECLAS LAS FÓRMULAS SE PUEDEN INTRODUCIR DIRECTAMENTE.

POR EJEMPLO, SE REQUIERE UNA MACRO QUE MULTIPLIQUE LOS VALORES ALMACENADOS EN LAS CELDAS IMPORTE E IMP Y ALMACENAR EL RESULTADO

EN LA CELDA TOTAL. ESTO SE PUEDE REALIZAR YA SEA CON EL COMANDO {LET} O CON UNA SECUENCIA DE TECLAS. AUNQUE, EN AMBOS CASOS EL DESPLIEGUE DEL RESULTADO ES EL MISMO, EL CONTENIDO DE LA CELDA ES DISTINTO. EN EL CASO DEL COMANDO EL VALOR ES FIJO Y EN EL CASO DE LA SECUENCIA CADA VEZ QUE VARIE EL VALOR DE IMPORTE O IMP, TAMBIÉN VARIARÁ LA CELDA TOTAL.

{LET PAGO IMPORTE*IMP}

CONTENIDO		DESPLIEGUE	
IMPORTE	10000	IMPORTE	10000
IMP	0.08	IMP	0.08
TOTAL	800	TOTAL	800

{GOTO}PAGO~+IMPORTE*IMP~

CONTENIDO		DESPLIEGUE	
IMPORTE	10000	IMPORTE	10000
IMP	0.08	IMP	0.08
TOTAL	+IMPORTE*IMP	TOTAL	800

EL COMANDO {LET} INTRODUCE LOS VALORES CALCULADOS NO LAS FÓRMULAS

UNA IMPORTANTE UTILIZACIÓN DEL COMANDO {LET} ES EL MANEJO DE CONTADORES, ACUMULADORES O APUNTADES A VARIABLES, DADO QUE ESTO SE UTILIZA EN PROCESOS REPETITIVOS. PARA MANTENER ESTE TIPO DE VARIABLES SE PUEDE UTILIZAR EL COMANDO {LET}.

```

+-----+-----+
| CONT   | 11 | {LET CONT,CONT+1} --> | CONT   | 12 |
+-----+-----+

```

IMPLEMENTACIÓN DE UN CONTADOR VARIABLE CON {LET}

UNA MACRO TAMBIÉN PUEDE BORRAR LA INFORMACIÓN DE UNA CELDA. EL COMANDO {BLANK} ES UTILIZADO PARA ESTA FUNCIÓN. DE LA MISMA MANERA SE PUEDE UTILIZAR LA SECUENCIA DE TECLAS /RANGE ERASE. EL COMANDO {BLANK}, SIN EMBARGO, TIENE VARIAS VENTAJAS. UNA DE ELLAS ES EL PODER EJECUTARSE EN CUALQUIER MOMENTO, OTRA ES QUE LA VELOCIDAD ES MAYOR, DADO QUE AL EJECUTAR UN COMANDO, LA PANTALLA NO SE ACTUALIZA NI SE EJECUTA UN RECÁLULO DE LA HOJA.

MANEJO DE TABLAS.-

MUCHAS APLICACIONES DE PROCESAMIENTO DE DATOS INVOLUCRAN INFORMACIÓN QUE ES MÁS FÁCIL MANEJAR COMO TABLAS, EN LUGAR DE CELDAS INDIVIDUALES. POR EJEMPLO, CUANDO SE TIENEN CIERTOS CATÁLOGOS, SE BUSCA LA INFORMACIÓN EN UNA DETERMINADA COLUMNA Y SE REQUIERE LA INFORMACIÓN DE OTRA COLUMNA PERO DEL MISMO RENGLÓN.

EN 1-2-3 EL PROCESO DE MANEJAR TABLAS ES MUY SENCILLO. EN UNA MACRO SE PUEDE UTILIZAR LA COMBINACIÓN DE LAS FUNCIONES @VLOOKUP, @HLOOKUP Y @INDEX Y EL COMANDO {PUT} PARA PROCESAR CUALQUIER RANGO DE DATOS -POR EJEMPLO, UN GRUPO RECTANGULAR DE CELDAS- COMO UNA TABLA.

LOCALIZACIÓN DE UN DATO DENTRO DE UNA TABLA.-

PARA LOCALIZAR UN DATO DENTRO DE UNA TABLA SE TENDRÁ QUE HACER REFERENCIA A TRES ELEMENTOS: LA LOCALIZACIÓN DE LA TABLA DENTRO DE LA HOJA, LA COLUMNA DONDE SE LOCALIZA EL DATO Y EL RENGLÓN DONDE SE LOCALIZA EL DATO. LA COLUMNA Y EL RENGLÓN, A MENUDO SON ESPECIFICADOS CON UN DETERMINADO DESPLAZAMIENTO (OFFSET). EL DESPLAZAMIENTO SE REFIERE A LA POSICIÓN RELATIVA QUE GUARDA LA COLUMNA Y EL RENGLÓN CON RESPECTO A LA CELDA SUPERIOR IZQUIERDA DE LA TABLA, CUYO VALOR ES COLUMNA 0 RENGLÓN 0. POR EJEMPLO, UNA COLUMNA CON DESPLAZAMIENTO 4 SE REFIERE A LA QUINTA COLUMNA, Y UN RENGLÓN CON DESPLAZAMIENTO 7 SE REFIERE AL OCTAVO RENGLÓN.

RECUPERANDO INFORMACIÓN DE UNA TABLA.-

1-2-3 TIENE TRES FUNCIONES PARA RECUPERAR INFORMACIÓN DE UNA TABLA: @VLOOKUP, @HLOOKUP Y @INDEX. LAS DOS PRIMERAS FUNCIONES REALIZAN LA RECUPERACIÓN CON RESPECTO AL DATO QUE SE QUIERE LOCALIZAR -EL PROCESO DE RECUPERACIÓN DEL DATO INICIA CON LA COMPARACIÓN DE UN VALOR SELECCIONADO DENTRO DE LA PRIMERA COLUMNA O RENGLÓN DE LA TABLA-. LA FUNCIÓN @INDEX, POR EL OTRO LADO, REALIZA LA BÚSQUEDA DEPENDIENDO ESTRICTAMENTE DE UNA POSICIÓN DADA -LA INFORMACIÓN SE RECUPERA DANDO LOS DESPLAZAMIENTOS DE LA COLUMNA Y EL RENGLÓN DENTRO DE LA TABLA-.

@HLOOKUP("TEL", TABLA, 2)

	A	B	C	D	E	F
1	TABLA	AGUA	LUZ	PREDIO	TEL	
2	ZONA 1	5	7	8	6	10
3	ZONA 2	3	9	15	9	12
4	ZONA 3	7	11	22	15	15

- A) IDENTIFICA LA TABLA (RANGO TABLA, B1..F4).
- B) USA EL VALOR SELECCIONADO (TEL) PARA REALIZAR LA BÚSQUEDA DENTRO DEL PRIMER RENGLÓN.
- C) USA EL DESPLAZAMIENTO DE RENGLÓN (2) PARA LOCALIZAR LA CELDA CORRESPONDIENTE DENTRO DE LA COLUMNA.
- D) DA EL VALOR SELECCIONADO DENTRO DE LA TABLA DANDO COMO RESULTADO 9 DENTRO DEL @HLOOKUP.

LA FUNCIÓN @VLOOKUP RECUPERA INFORMACIÓN, PERO REALIZA LA BÚSQUEDA EN LA PRIMERA COLUMNA DE LA TABLA Y DESPUÉS HACE EL DESPLAZAMIENTO.

LO SIGUIENTE MUESTRA LA FUNCIÓN @INDEX

@INDEX(TABLA,3,2)

	A	B	C	D	E	F
1	TABLA	AGUA	LUZ	PREDIO	TEL	
2	ZONA 1	5	7	8	6	10
3	ZONA 2	3	9	15	9	12
4	ZONA 3	7	11	22	15	15

- A) IDENTIFICA LA TABLA (RANGO TABLA, B1..F4).
- B) USA EL DESPLAZAMIENTO DE COLUMNA (3) PARA LOCALIZAR LA COLUMNA DETERMINADA.
- C) USA EL DESPLAZAMIENTO DE RENGLÓN (2) PARA LOCALIZAR LA CELDA CORRESPONDIENTE DENTRO DE LA COLUMNA.
- D) DA EL VALOR SELECCIONADO DENTRO DE LA TABLA DANDO COMO RESULTADO 9 DENTRO DEL @INDEX.

EL @INDEX, EL @HLOOKUP Y EL @VLOOKUP EXTRAEN VALORES DE UNA TABLA

INTRODUCIENDO DATOS A UNA TABLA.-

PARA INTRODUCIR DATOS A UNA TABLA SE UTILIZARÁ EL COMANDO {PUT}. SU FUNCIONAMIENTO SE DIVIDE EN DOS PARTES, LOCALIZAR LA POSICIÓN DENTRO DE LA TABLA DONDE INTRODUCIRÁ EL DATO E INTRODUCIR EL DATO -SE PUEDE DECIR QUE HACE LAS FUNCIONES DE MANERA PARALELA AL @INDEX Y AL COMANDO {LET} RESPECTIVAMENTE-. DE

MANERA SEMEJANTE A LA FUNCIÓN @INDEX, EL COMANDO {PUT} LOCALIZA LA CELDA DONDE INTRODUCIRÁ EL DATO USANDO EL DESPLAZAMIENTO DE LA COLUMNA Y EL RENGLÓN. Y TAMBIÉN DE MANERA SEMEJANTE AL COMANDO {LET}, INTRODUCE LAS ETIQUETAS ALINEADAS A LA IZQUIERDA, Y LOS VALORES NUMÉRICOS ALINEADOS A LA IZQUIERDA.

{PUT TABLA,4,0,"OTROS"}

	A	B	C	D	E	F
1	TABLA	AGUA	LUZ	PREDIO	TEL	OTROS
2	ZONA 1	5	7	8	6	10
3	ZONA 2	3	9	15	9	12
4	ZONA 3	7	11	22	15	15

- IDENTIFICA LA TABLA (RANGO TABLA, B1..F4)
- USA EL DESPLAZAMIENTO DE COLUMNA (4) PARA LOCALIZAR LA COLUMNA DETERMINADA.
- USA EL DESPLAZAMIENTO DE RENGLÓN (0)
- COLOCA LA ETIQUETA "OTROS" EN LA INTERSECCIÓN DE LA COLUMNA Y EL RENGLÓN LOCALIZADOS.

CONVERSIÓN ENTRE NÚMEROS Y LETRAS.-

CUALQUIER DATO QUE SE INTRODUZCA DENTRO DE UNA CELDA SERÁ DEL TIPO NUMÉRICO O DEL TIPO STRING. 1-2-3 COMO MUCHOS LENGUAJES DE PROGRAMACIÓN HACE UNA DIFERENCIA ENTRE ESTOS DOS TIPOS DE DATOS. CUANDO SE REQUIERAN REALIZAR ALGUNOS CÁLCULOS SERÁ CONVENIENTE REALIZAR LAS CONVERSIONES NECESARIAS, YA QUE DE OTRA MANERA SE PUEDEN PROVOCAR ERRORES O CÁLCULOS INCORRECTOS.

CON LAS MACROS SE PUEDEN REALIZAR LAS CONVERSIONES NECESARIAS UTILIZANDO EL COMANDO {CONTENTS} Y LAS FUNCIONES @STRING, @VALUE, @DATEVALUE Y @TIMEVALUE.

CONVERSIONES DE NÚMERO A STRING.-

NORMALMENTE, UNA CELDA COMPUESTA DE DÍGITOS CONTIENE UN VALOR NUMÉRICO. MUCHOS PROCESOS DE CÁLCULO REQUIEREN VALORES NUMÉRICOS, SIN EMBARGO, EN OTRAS OCASIONES SE REQUIERE UN VALOR DEL TIPO

STRING. POR EJEMPLO, SI REQUIERE MANDAR UN MENSAJE DEL SIGUIENTE TIPO: "SE TIENE UN SALDO DE \$350,000 ". SE TENDRÁ QUE CONVERTIR PRIMERO EL VALOR NUMÉRICO A STRING.

LA FUNCIÓN @STRING PUEDE SER UTILIZADA DENTRO DE UNA MACRO PARA REALIZAR LA CONVERSIÓN DE NÚMERO A STRING. ESTA FUNCIÓN FORMATEA EL NÚMERO, UTILIZANDO EL FORMATO FIXED Y UN NÚMERO ESPECIFICADO DE POSICIONES DECIMALES (DE 0 A 15). CONVIRTIENDO EL NÚMERO FORMATEADO A ETIQUETA. DEPENDIENDO DEL NÚMERO DE DECIMALES ESPECIFICADO, ESTE SERÁ REDONDEADO.

{LET DEST,@STRING(FUENTE,0)}

B1: (C2) 287.509	B2: '288
A B	A B
1 FUENTE \$287.51	2 DEST 288

CONVERSIÓN DE NÚMERO A STRING CON @STRING

EN ESTE CASO LA FUNCIÓN @STRING CONVIERTE A ETIQUETA EL CONTENIDO DE LA CELDA FUENTE Y LO COLOCA EN LA CELDA DEST REDONDEANDO A CERO DECIMALES.

EL COMANDO {CONTENTS} TAMBIÉN CONVIERTE UN NÚMERO A UNA ETIQUETA. DE MANERA DIFERENTE A LA FUNCIÓN @STRING, {CONTENTS} PUEDE REFORMATAR UN NÚMERO A CUALQUIER FORMATO VÁLIDO DE 1-2-3.

{CONTENTS DEST,FUENTE}

B1: (C2) 287.509	B2: ' \$287.51
A B	A B
1 FUENTE \$287.51	2 DEST \$287.51

CONVERSIÓN DE NÚMERO A STRING CON {CONTENTS}

LO ANTERIOR PERMITE QUE LA APARIENCIA DEL NÚMERO NO SEA MODIFICADA. SE PUEDE ESPECIFICAR ADICIONALMENTE EL ANCHO DE LA CELDA DONDE SE DEPOSITARÁ LA ETIQUETA, POR EJEMPLO, PARA CENTRARLA. TAMBIÉN SE PUEDE ESPECIFICAR EL FORMATO QUE PUEDE SER CUALQUIERA DE LOS SIGUIENTES VALORES:

0	-	15	FIJO DE 0 A 15 DECIMALES
16	-	32	NOTACIÓN CIENTÍFICA DE 0 A 15 DECIMALES
32	-	47	NOTACIÓN CON \$ Y DE 0 A 15 DECIMALES
48	-	63	PORCENTAJE DE 0 A 15 DECIMALES
64	-	79	PUNTUAR DE 0 A 15 DECIMALES
113			GENERAL
114			(DDD-MMM-AA)

115 (DD-**MMM**)
 116 (**MMM-AA**)
 121 (**MM/DD/AA**) VARÍA SEGÚN SE HAYA CONFIGURADO EN EL
 SETTINGS
 122 (**MM/DD**) VARÍA SEGÚN SE HAYA CONFIGURADO EN EL
 SETTINGS
 119 (**HH:MM:SS AM/PM**)
 120 (**HH:MM AM/PM**)
 123 (**HH:MM:SS**) VARÍA SEGÚN SE HAYA CONFIGURADO EN EL
 SETTINGS
 124 (**HH:MM**) VARÍA SEGÚN SE HAYA CONFIGURADO EN EL
 SETTINGS
 117 LITERAL (DUPLICA EL CONTENIDO DE FUENTE)
 118 OCULTO
 127 FORMATO DE DEFAULT NUMÉRICO

CONVERSIONES DE STRING A NÚMERO. -

EL ALGUNAS SITUACIONES, UNA SERIE DE DÍGITOS SON INTRODUCIDOS DENTRO DE LA HOJA DE CÁLCULO COMO UNA ETIQUETA O PARTE DE UNA ETIQUETA EN LUGAR DE UN NÚMERO. ESTO SUCEDE POR EJEMPLO, CUANDO SE QUIERE INCORPORAR UN DOCUMENTO DENTRO DE 1-2-3 QUE FUE CREADO CON UN PROCESADOR DE TEXTOS. O CUANDO SE TIENE UN CONJUNTO DE VALORES QUE FUERON INTRODUCIDOS COMO ETIQUETAS POR ERROR.

PARA PODER UTILIZAR ESTE STRING DE DÍGITOS PARA CIERTOS CÁLCULOS O PARA PODER SER GRAFICADOS, DEBERÁN SER CONVERTIDOS A NÚMEROS. LAS FUNCIONES QUE PERMITEN REALIZAR ESTA TAREA SON @VALUE, @DATEVALUE Y @TIMEVALUE. PARA ESTE CASO NO SE CUENTA CON UN COMANDO DE PROGRAMACIÓN.

LA FUNCIÓN @VALUE CONVIERTE UN STRING DE DÍGITOS EN NÚMEROS. LA FUNCIÓN NO TOMA EN CUENTA SI EXISTEN ESPACIOS A LA DERECHA O A LA IZQUIERDA DE LOS DÍGITOS.

{LET DEST,@VALUE(FUENTE)}

B1: ' 287.5 B
 A
 +-----+
 1| FUENTE | 287.5 |
 +-----+

B2: (C1) 287.5
 A B
 +-----+
 2| DEST | \$287.5 |
 +-----+

CONVERSIÓN DE STRING A NÚMERO CON @VALUE

A DIFERENCIA DEL COMANDO {?} QUE PERMITE DEJAR EL CONTROL AL USUARIO, EXISTEN CUATRO INSTRUCCIONES QUE ALMACENAN LA INFORMACIÓN EN UNA CELDA DETERMINADA. LA INFORMACIÓN VARÍA DEPENDIENDO DEL COMANDO, PUDIENDO SER ETIQUETAS O NÚMEROS, DE UN SOLO CARACTER O DE VARIOS.

LA INSTRUCCIÓN {GETLABEL} PERMITE INTRODUCIR UNA ETIQUETA DADA POR EL USUARIO A LA CELDA ESPECIFICADA. CON ESTE COMANDO SE PUEDE DESPLEGAR EN EL PANEL DE CONTROL UN MENSAJE QUE FACILITE AL USUARIO SABER QUE INFORMACIÓN DEBERÁ INTRODUCIR.

A) DESPLIEGA EL MENSAJE EN EL PANEL DE CONTROL, LO QUE DIGITA EL USUARIO ES COLOCADO EN LA CELDA ARC

B) LA INFORMACIÓN PRES ES COLOCADA EN ARC, EL FILE COMBINE SE REALIZA SOBRE EL ARCHIVO PRES.

```

+-----+-----+
| INICIA | {GETLABEL ARCHIVO? ,ARC} |
+-----+-----+
|         | {HOME}/FCC~n             |
+-----+-----+
| ARC +->|
+-----+-----+
|         | ~                             |
+-----+-----+
|         | ARCHIVO?                         |
| P       | +-----+-----+           |
| R       | | TECLADO | | <-----+
| E       | |         | | <--+
| S       | +-----+-----+           |
+-----+-----+

```

EL COMANDO {GETLABEL} PERMITE PROCESAR ETIQUETAS

DE IGUAL FORMA QUE EN {GETLABEL} EL COMANDO {GETNUMBER} PERMITE INTRODUCIR INFORMACIÓN A UNA CELDA, PERO DE TIPO NUMÉRICO. TAMBIÉN SE PUEDE DESPLEGAR UN MENSAJE EN EL PANEL DE CONTROL. AMBAS INSTRUCCIONES PUEDEN SER DE BASTANTE UTILIDAD DADO QUE PERMITEN DESPLEGAR MENSAJES QUE DARÁN CLARIDAD AL USUARIO. DENTRO DE ESTOS MENSAJES SE PUEDE, POR EJEMPLO, DEFINIR RÁNGOS VÁLIDOS EN LOS DATOS O DATOS VÁLIDOS.

A) DESPLIEGA EL MENSAJE EN EL PANEL DE CONTROL, LO QUE DIGITA EL USUARIO ES COLOCADO EN LA CELDA NUM

B) EL NÚMERO 5 ES COLOCADO EN LA CELDA NUM. EL COMANDO {FOR} SE REALIZA EL NÚMERO DE VECES ESPECIFICADO EN NUM.

```

+-----+-----+
| INICIA | {GETNUMBER COPIAS? ,NUM} |
+-----+-----+
|         | {FOR I,1,NUM,1,IMPRIME}         |
+-----+-----+
| NUM +->|
+-----+-----+
|         | COPIAS?                         |
| 5 |     | +-----+-----+           |
|   |     | | TECLADO | | <-----+
|   |     | |         | | <--+
|   |     | +-----+-----+           |
+-----+-----+

```

EL COMANDO {GETNUMBER} PERMITE PROCESAR NÚMEROS

EN OCASIONES ES NECESARIO REALIZAR UNA PAUSA PARA DIGITAR UN SOLO CARACTER DESDE EL TECLADO, EL COMANDO {GET} REALIZA LA PAUSA HASTA QUE SE DIGITE CUALQUIER TECLA. POR EJEMPLO, LETRA, DÍGITO, TECLAS DE FUNCIÓN, TECLAS DE DESPLAZAMIENTO, RETURN, ETC. ESTE COMANDO NO DESPLIEGA MENSAJE EN EL PANEL DE CONTROL, SOLO ESPERA UN CARACTER Y LO ALMACENA EN LA CELDA QUE SE INDIQUE.

```
+-----+
| {GET RESPUESTA} |
+-----+
```

EN CONTRASTE AL COMANDO {GET}, EL CUAL SUSPENDE LA EJECUCIÓN Y DA EL CONTROL AL USUARIO, EL COMANDO {LOOK} PROCESA EL CARACTER QUE SE DIGITE MIENTRAS LA MACRO ESTÁ EN EJECUCIÓN. AUNQUE NO EXISTE UNA PAUSA DURANTE EL COMANDO {LOOK}, LA MACRO EJECUTA DOS COSAS SIMULTÁNEAMENTE: EJECUTA LA INSTRUCCIÓN Y DEJA "READY" EL TECLADO PARA PROCESAR EL SIGUIENTE CARACTER QUE SE DIGITE.

ESTE COMANDO ES ÚTIL CUANDO SE REQUIERE PARAR UN PROCESO SIN NECESIDAD DE REALIZAR PAUSAS PARA QUE SE SOLICITE SU TERMINACIÓN, ESTO SE PUEDE DADO QUE EL ÚLTIMO CARACTER DIGITADO QUEDA ALMACENADO EN LA CELDA ESPECIFICADA, PUDIENDO PREGUNTAR POR ÉL EN CUALQUIER MOMENTO.

ALGUNAS APLICACIONES INTERACTIVAS EN OCASIONES REQUIEREN QUE SEAN SUSPENDIDAS DURANTE UN TIEMPO DETERMINADO Y POSTERIORMENTE CONTINUEN AUTOMÁTICAMENTE CON EL PROCESO. EL COMANDO {WAIT} PERMITE SUSPENDER UNA MACRO DURANTE UN TIEMPO DETERMINADO.

LA EJECUCIÓN CONTINUA EN FUNCIÓN DEL TIEMPO ESPECIFICADO POR EL ARGUMENTO DEL COMANDO. POR EJEMPLO, SE PUEDE UTILIZAR PARA MANDAR UN MENSAJE, DURANTE UN TIEMPO SUFICIENTE PARA QUE EL USUARIO LO LEA. POSTERIORMENTE LA MACRO PODRÁ CONTINUAR.

EL COMANDO {WAIT} NO SOLO SE APLICA EN PROGRAMAS INTERACTIVOS, SINO TAMBIÉN EN PROGRAMAS QUE NECESITAN EJECUTARSE EN DETERMINADOS HORARIOS.

EL PROCESO SE SUSPENDE
5 SEGUNDOS

```
+-----+
| {WAIT @NOW+@TIME(0,0,5)} |
+-----+
```

FUNCIONES @CELLPOINTER Y @CELL

LAS FUNCIONES @CELLPOINTER Y @CELL PERMITEN CONOCER CIERTA INFORMACIÓN SOBRE LA CELDA APUNTADA, COMO POR EJEMPLO:

ADDRESS - MUESTRA LA DIRECCIÓN ABSOLUTA DE LA CELDA

- COL - MUESTRA EL NÚMERO DE COLUMNA QUE OCUPA LA CALDA (1 - 256)
- CONTENTS - MUESTRA EL CONTENIDO DE LA CELDA
- FORMAT - MUESTRA EL FORMATO DE LA CELDA
- PREFIX - MUESTRA EL PREFIJO QUE HAY EN LA CELDA ('', ^, \, |)
- PROTECT - ARROJA UN 0 SI LA CELDA ESTA DESPROTEGIDA O UN 1 SI LA CELDA ESTA PROTEGIDA
- ROW - MUESTRA EL NÚMERO DE RENGLÓN QUE OCUPA LA CELDA (1 - 8192)
- TYPE - MUESTRA EL TIPO DE INFORMACIÓN QUE TIENE LA CELDA
 - B - CELDA VACÍA
 - L - CELDA CON UNA ETIQUETA
 - V - CELDA CON UN VALOR NUMÉRICO O UNA FÓRMULA

@CELLPOINTER("CONTENTS")

	A	B
1	MES	ENERO

ARROJA COMO RESULTADO LA ETIQUETA ENERO SI EL APUNTADOR ESTA EN B1

LA FUNCIÓN @CELL ARROJA LA INFORMACIÓN DE FORMA SIMILAR PERO HACIENDO REFERENCIA A UN RANGO DE UNA CELDA.

@CELL("TYPE",MES)

MES	ENERO
-----	-------

EL RESULTADO ES LA ETIQUETA L, PORQUE EL TIPO DE INFORMACIÓN QUE CONTIENE LA CELDA MES ES UNA ETIQUETA.

VI RECALCULACION Y REDESPLIEGUE

EL INTRODUCIR INFORMACIÓN A UNA CELDA O BORRARLA GENERALMENTE CAMBIA EL VALOR DE LA CELDA EN CUESTIÓN Y TODAS AQUELLAS CELDAS QUE CONTIENEN FÓRMULAS DEPENDIENTES. CUANDO UNA SECUENCIA DE

TECLAS CREA O BORRA UNA CELDA, 1-2-3 ACTUALIZA LA HOJA DE CÁLCULO Y DESPLIEGA LOS CAMBIOS AUTOMÁTICAMENTE (SIEMPRE Y CUANDO EL MÉTODO DE RECALCULACIÓN ESTE EN AUTOMÁTICO). CUANDO UN COMANDO DE PROGRAMACIÓN CREA O BORRA UNA CELDA, 1-2-3 NO ACTUALIZA AUTOMÁTICAMENTE LA HOJA NI LA PANTALLA. PARA REALIZARLO SE TENDRÁN QUE INVOCAR LOS COMANDO {CALC} O SIMPLEMENTE INCLUIR RETURN A LA MACRO. O CAMBIAR DE POSICIÓN DENTRO DE LA CELDA.

EL COMANDO {CALC} RECALCULA LAS FÓRMULAS DE TODA LA HOJA Y ACTUALIZA O REDESPLIEGA LA VENTANA.

HAY OTROS COMANDOS QUE TAMBIÉN SON PARA EL DESPLEGADO Y SON: {RECALC} Y {RECALCCOL}.

EL COMANDO {RECALC} RECALCULA TODAS LAS FÓRMULAS DE UN RANGO ESPECIFICADO RENGLÓN POR RENGLÓN.

EL COMANDO {RECALCCOL} RECALCULA TODAS LAS FÓRMULAS DE UN RANGO ESPECIFICADO COLUMNA POR COLUMNA.

CONTROLANDO EL DESPLEGADO EN LA PANTALLA.

EN LA EJECUCIÓN DE UNA MACRO PUEDE CONTROLARSE EL DESPLEGADO DE LA HOJA Y DEL PANEL DE CONTROL PARA DAR UNA MEJOR PRESENTACIÓN. POR EJEMPLO SI SE TIENE LA SIGUIENTE MACRO

```
+-----+-----+
| \B                | '/RFC2~{END}{DOWN}~ |
+-----+-----+
```

- / DESPLIEGA EL MENÚ EN EL PANEL DE CONTROL, CAMBIA EL INDICADOR A MENU
- R DESPLIEGA OTRO MENÚ
- F DESPLIEGA OTRO MENÚ
- C DESPLIEGA EL PROMPT "NUMBER OF DECIMAL PLACES.... (0..15)" EN EL PANEL DE CONTROL Y CAMBIA EL MODO DEL INDICADOR A EDIT
- 2 DESPLIEGA EL NÚMERO 2 EN EL PANEL DE CONTROL
- ~ DESPLIEGA EL PROMPT "RANGE TO FORMAT..." EN EL PANEL DE CONTROL Y CAMBIA EL MODO DEL INDICADOR A POINT

{END} EL CURSOR SE MUEVE AL FINAL DE LA HOJA

{DOWN} SENALA LA COLUMNA DE CELDAS

~ QUITA LA MARCA (ILUMINADO), LIMPIA EL PANEL DE CONTROL Y CAMBIA EL MODO DEL INDICADOR A READY

EN ESTA SECCIÓN SE VERÁN LOS COMANDOS {PANELOFF} Y {PANELON} PARA CONTROLAR EL DESPLEGADO DE LOS MENSAJES EN EL PANEL DE CONTROL, {WINDOWSOFF} Y {WINDOWSON} PARA CONTROLAR EL DESPLEGADO EN LA HOJA, {INDICATE} PARA CAMBIAR MODO DEL INDICADOR Y {BEEP} PARA HACER UNA LLAMADA DE ATENCIÓN.

```
+-----+-----+
| \D          | {WINDOWSOFF}{PANELOFF} |
+-----+-----+
|             | '/RFC2~{END}{DOWN}~   |
+-----+-----+
|             | {PANELON}{WINDOWSON}  |
+-----+-----+
```

EJECUTANDO ESTA MACRO SE EVITARÁ QUE EN EL PANEL DE CONTROL SE DESPLIEGUEN LOS MENÚS Y LOS PROMPT Y QUE NO SE VEA EL SENALAMIENTO EN LA HOJA.

CON LOS COMANDOS {PANELON} Y {WINDOWSON} AL FINAL DE LA MACRO, SE VUELVEN A ACTIVAR EL DESPLEGADO DEL PANEL DE CONTROL Y DE LA HOJA.

EL COMANDO {INDICATE} CAMBIA EL MODO DEL INDICADOR POR EL QUE SE ESPECIFIQUE.

```
+-----+-----+ A) CAMBIA EL MODO DEL
| \F          | {INDICATE "RUTIN"} | INDICADOR POR RUTIN
+-----+-----+
|             | {RUTINA}           | B) LLAMA LA SUBROUTINA
+-----+-----+ RUTINA
|             | {INDICATE}         | C) RESTABLECE LA OPERACIÓN
+-----+-----+ ESTÁNDAR DEL INDICADOR
```

LA CADENA QUE SE UTILICE PARA INDICAR EL MODO PUEDE SER HASTA DE 5 CARACTERES.

EL COMANDO {BEEP} PRODUCE UN SONIDO PARA HACER UNA LLAMADA DE ATENCIÓN CUANDO SE REQUIERA DURANTE LA EJECUCIÓN DE LA MACRO, POR EJEMPLO CUANDO SE TIENE UNA MACRO INTERACTIVA Y SE REQUIERE QUE EL USUARIO PROPORCIONE ALGUNA INFORMACIÓN.


```
+-----+  
| \g      | 'WC{BEEP}S{?}~ |  
+-----+
```

LA EJECUCIÓN DE ESTA MACRO OCASIONA QUE CUANDO SE REQUIERE QUE EL USUARIO PROPORCIONE EL TAMAÑO AL QUE DEBERÁ MODIFICARSE LA COLUMNA, SE PRODUZCA UN SONIDO PARA LLAMAR SU ATENCIÓN.

EL COMANDO {BEEP} PUEDE UTILIZARSE CON LOS SIGUIENTES VALORES:

- {BEEP 0} - PRODUCE UN TONO MUY BAJO
- {BEEP 1} - PRODUCE UN TONO BAJO
- {BEEP 2} - PRODUCE UN SONIDO ALTO
- {BEEP 3} - PRODUCE UN SONIDO MUY ALTO

VII MODIFICACION DEL CONTROL DEL FLUJO DEL PROGRAMA

UN PROCEDIMIENTO EN MACROS (RUTINA) PRODUCE EL MISMO RESULTADO CADA VEZ QUE ES INVOCADO. EN OTRAS PALABRAS LAS MACROS DE LAS RUTINAS SIEMPRE SE EJECUTAN EN LA MISMA SECUENCIA EN QUE FUERON DIGITADAS, DESDE EL INICIO HASTA EL FINAL. UNA RUTINA COMPLEJA, POR OTRO LADO, CONTIENE UN CONJUNTO DE INSTRUCCIONES QUE SON EJECUTADAS CUANDO UNA DETERMINADA CONDICIÓN SE CUMPLE. ESTO ES, EN FUNCIÓN DE DETERMINADAS CONDICIONES EL FLUJO DEL PROGRAMA SERÁ MODIFICADO PARA EJECUTAR CIERTO CÓDIGO. 1-2-3 DIRECCIONA EL FLUJO DEL PROGRAMA EN DOS FORMAS: POR MEDIO DE SALTOS (GOTO'S) O POR MEDIO DE LLAMADAS A SUBRUTINAS.

CUANDO UNA MACRO SALTA, EL CONTROL CONTINUA EN LA NUEVA LOCALIDAD Y LA EJECUCIÓN PROSIGUE AHÍ. EL FLUJO DEL PROGRAMA NO REGRESA AUTOMÁTICAMENTE CUANDO EL CONJUNTO DE INSTRUCCIONES ES EJECUTADA. UNA SUBROUTINA TAMBIÉN LLAMA A UNA LOCALIDAD ESPECÍFICA PARA REALIZAR UN CONJUNTO DE INSTRUCCIONES, PERO A DIFERENCIA DE LOS SALTOS, EL CONTROL DEL PROGRAMA REGRESA A LA SIGUIENTE LOCALIDAD A PARTIR DE DONDE SE LLAMÓ.

ESTA SECCIÓN DEL MANUAL DISCUTE LAS TÉCNICAS PARA MODIFICAR EL FLUJO DEL PROGRAMA POR MEDIO DE SALTOS. LOS COMANDOS UTILIZADOS PARA ELLO SON: {BRANCH}, {IF}, {DISPATCH} Y {QUIT}. POSTERIORMENTE EN OTRA SECCIÓN SE DISCUTIRÁN LAS LLAMADAS A SUBRUTINAS.

SALTO DIRECTO.-

EL COMANDO {BRANCH} TIENE DOS OBJETIVOS PRINCIPALES. EL PRIMERO ES EL DE ORGANIZAR - ESTO ES, HACER EL CÓDIGO MÁS ACCESIBLE PARA EL PROGRAMADOR, PROBABLEMENTE BIEN UTILIZADO PUEDE HACER EL CÓDIGO MÁS LEGIBLE-. CUANDO SE DESARROLLA EN MUCHAS OCASIONES EL CÓDIGO GENERADO NO ALCANZA A DESPLEGARSE DENTRO DE UNA COLUMNA, PARA ELLO SE PUEDE CAMBIAR EL FLUJO PARA CREAR PROGRAMAS EN DOS O MÁS COLUMNAS.

EL COMANDO {BRANCH} TAMBIÉN PUEDE SER UTILIZADO PARA MODIFICAR EL FLUJO DEL PROGRAMA EN FUNCIÓN DE UNA CONDICIÓN DETERMINADA. EN ESTOS CASOS EL COMANDO {IF} PRECEDERÁ AL COMANDO {BRANCH}. SI LA CONDICIÓN EVALUADA DENTRO DEL {IF} ES VERDADERA, EL FLUJO DEL PROGRAMA CONTINUARÁ EN LA LOCALIDAD QUE EL COMANDO {BRANCH} INDIQUE. SI ES FALSA LA EVALUACIÓN, LA EJECUCIÓN CONTINUARÁ SECUENCIALMENTE.

RUTINA	{IF CONT>10}{BRANCH IMPRIME}	(A)
	{LET CONT,CONT+1}	(B)
	{PROCESA}	
	{BRANCH RUTINA}	

(A) SI LA CONDICIÓN DEL {IF} (CONT>10) ES VERADERA, LA MACRO SALTA A LA RUTINA IMPRIME.

(B) SI LA CONDICIÓN DEL {IF} (CONT>10) ES FALSA CONTINUA EL PROCESO SECUENCIALMENTE EJECUTANDO EL {LET}.

IMPLEMENTACIÓN DE UN SALTO CONDICIONAL

SALTO INDIRECTO.-

NO TODAS LAS SITUACIONES SON SEMEJANTES AL EJEMPLO ANTERIOR, DE SOLO MANEJAR DOS POSIBLES RESULTADOS. POR EJEMPLO, LA FÓRMULA SALDO + DEPOSITO PODRÁ SER MENOR QUE, IGUAL A O MAYOR QUE EL VALOR DEPOSITADO EN LA CELDA PAGO, Y CADA UNO DE ESTOS TRES POSIBLES RESULTADOS REQUERIRÁ DE UN PROCESO DIFERENTE.

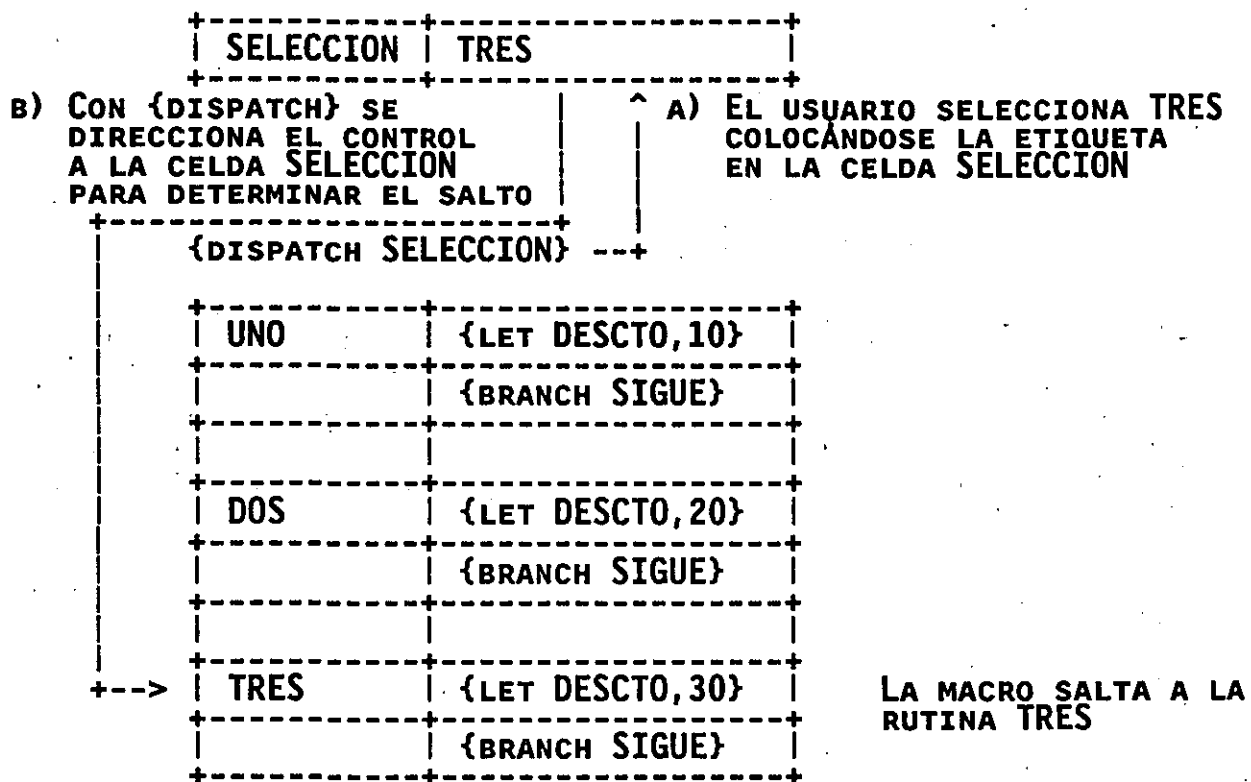
LOS COMANDOS {IF} Y {BRANCH} PUEDEN CONTROLAR SITUACIONES QUE INVOLUCREN CUALQUIER NÚMERO DE POSIBLES RESULTADOS. SIN EMBARGO, SE NECESITARÁ UN {IF} Y UN {BRANCH} SEPARADO PARA CADA POSIBLE RESULTADO. ESTO TRAERÁ POR CONSIGUIENTE, EL TENER QUE GENERAR UNA CANTIDAD MAYOR DE CÓDIGO. PARA RESOLVER ESTE PROBLEMA SE NECESITARÁ DE UNA SOLUCIÓN MÁS EFICIENTE, EL UTILIZAR EL COMANDO {DISPATCH}.

EL COMANDO {DISPATCH} PROPORCIONA LA FACILIDAD DE REALIZAR UN SALTO INDIRECTO DENTRO DEL FLUJO DEL PROGRAMA. EN OTRAS PALABRAS, EN LUGAR DE REDIRECCIONAR EL FLUJO DE LA MACRO A UNA NUEVA CELDA, EL COMANDO {DISPATCH} REDIRECCIONA EL CONTROL A LA CELDA QUE ESPECIFICARÁ LA NUEVA CELDA. LAS SIGUIENTES DOS MACROS REALIZAN LO MISMO (SE ASUME QUE EL USUARIO DESDE EL TECLADO DEPOSITARÁ EN LA CELDA SELECCION CUALQUIERA DE LAS ETIQUETAS UNO, DOS O TRES):

A) {IF SELECCION="UNO"}{BRANCH UNO}
 {IF SELECCION="DOS"}{BRANCH DOS}
 {IF SELECCION="TRES"}{BRANCH TRES}

B) {DISPATCH SELECCION}

LA SIGUIENTE FIGURA MUESTRA EL SALTO INDIRECTO QUE REALIZA EL SEGUNDO EJEMPLO:



EL COMANDO {DISPATCH} EJECUTA UN SALTO INDIRECTO

TERMINANDO UNA MACRO CON EL COMANDO {IF}.-

VAMOS A SUPONER QUE QUEREMOS PROCESAR UNA MACRO, SIEMPRE Y CUANDO UNA CONDICIÓN SEA FALSA. EN DICHO CASO SE PODRÁ UTILIZAR EL COMANDO {QUIT} PARA FINALIZAR LA MACRO CUANDO DICHA CONDICIÓN SEA VERDADERA. POR EJEMPLO:

```
{IF CONT=1000}{QUIT}
{BRANCH PROCESA}
```

ESTA SECUENCIA DE COMANDOS CAUSA QUE LA MACRO SEA TERMINADA CUANDO EL VALOR DE CONT ALCANCE EL VALOR DE 1000. NÓTESE QUE EL COMANDO {QUIT} DEBERÁ SEGUIR AL COMANDO {IF} PARA QUE LA MACRO FINALICE EN ESE PUNTO. SI EL VALOR DE CONT NO ES IGUAL A 1000, ENTONCES SE EJECUTARÁ LA RUTINA PROCESA.

VIII MANEJO DE SUBRUTINAS

EL LENGUAJE DE PROGRAMACIÓN DE 1-2-3 PERMITE DIVIDIR UN CONJUNTO GRANDE DE MACROS EN PEQUEÑOS GRUPOS O SUBRUTINAS. LA CAPACIDAD DE MANEJAR SUBRUTINAS ES COMÚN EN DIFERENTES LENGUAJES DE PROGRAMACIÓN TENIENDO DIFERENTES VENTAJAS.

LAS SUBRUTINAS HACEN MÁS FÁCIL LA ORGANIZACIÓN Y PROGRAMACIÓN DE UNA MACRO GRANDE, DADO QUE LAS MACRO-INSTRUCCIONES PUEDEN SER ESCRITAS EN DIFERENTES MÓDULOS. CADA UNO DE ESTOS MÓDULOS EJECUTA UNA FUNCIÓN EN PARTICULAR Y AL MOMENTO DE SER MÁS PEQUEÑAS, PERMITEN ENCONTRAR ERRORES DE MANERA MÁS FÁCIL.

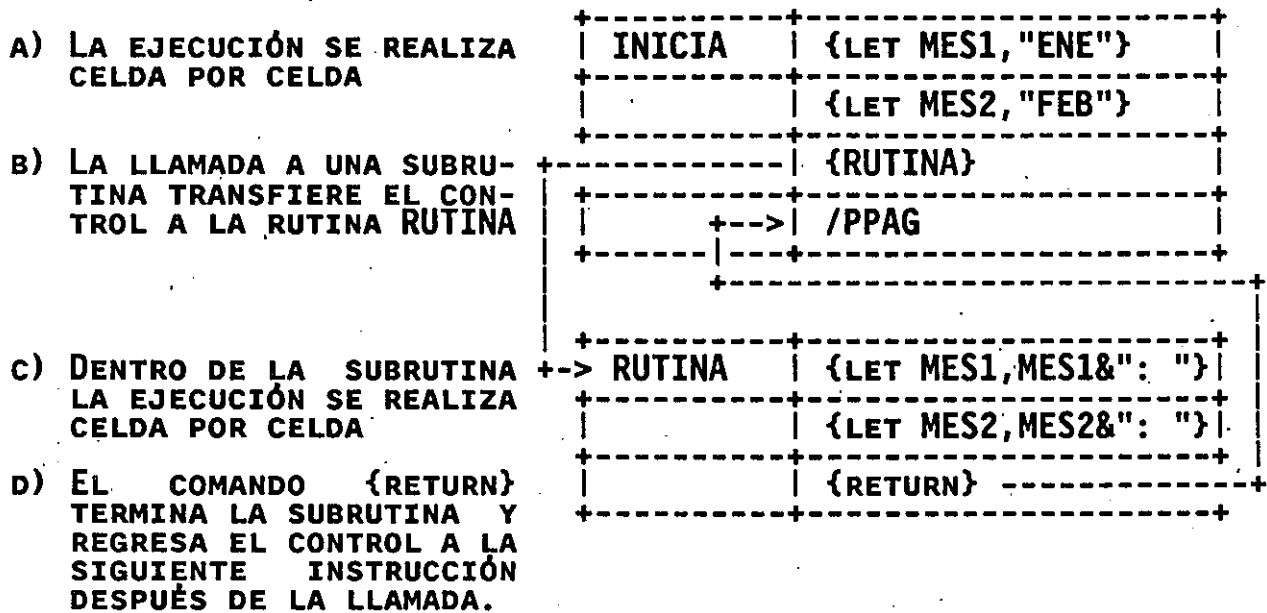
LAS SUBRUTINAS TAMBIÉN OFRECEN GRAN EFICIENCIA. EN MUCHAS APLICACIONES LA MISMA TAREA O TAREAS MUY SIMILARES SON REPETIDAS CONSTANTEMENTE. DIVIDIR UNA MACRO EN DIFERENTES SUBRUTINAS PERMITE EJECUTARLA EN DIFERENTES OCASIONES SIN TENER QUE REPETIR LA MACRO EN DIFERENTES PARTES DEL PROGRAMA. AL INTRODUCIR UNA VEZ EL CÓDIGO COMO SUBRUTINA, CUANDO SE NECESITEN EJECUTAR, SIMPLEMENTE SE INVOCARÁ A LA SUBRUTINA CON SU NOMBRE. ADEMÁS EL CORREGIR UN ERROR EN UNA SUBRUTINA O ADICIONAR UNA INSTRUCCIÓN ES MUCHO MÁS SENCILLO QUE EN UN PROGRAMA GRANDE.

FINALMENTE, INTRODUCIR UNA SECCIÓN DE CÓDIGO UNA SOLA VEZ HACE MÁS EFICIENTE EL USO DE LA MEMORIA.

EN ESTA SECCIÓN SE MENCIONARÁ COMO INICIAR Y TERMINAR UNA SUBRUTINA, EJECUTAR UNA SUBRUTINA UN DETERMINADO NÚMERO DE OCASIONES, PASAR VALORES A UNA SUBRUTINA Y ANIDAR SUBRUTINAS.

EJECUTAR UNA SUBRUTINA. -

NO EXISTE UN COMANDO PARA EJECUTAR UNA SUBRUTINA. SIMPLEMENTE SE COLOCA EL NOMBRE DE LA SUBRUTINA (NOMBRE DEL RANGO) DENTRO DE PARÉNTESIS DE LLAVE. CUANDO LA SUBRUTINA TERMINA SU EJECUCIÓN, EL CONTROL DEL PROGRAMA REGRESA A LA INSTRUCCIÓN QUE INMEDIATAMENTE SIGUE A LA LLAMADA DE LA SUBRUTINA.



LA EJECUCIÓN DE UNA SUBROUTINA EN UNA MACRO

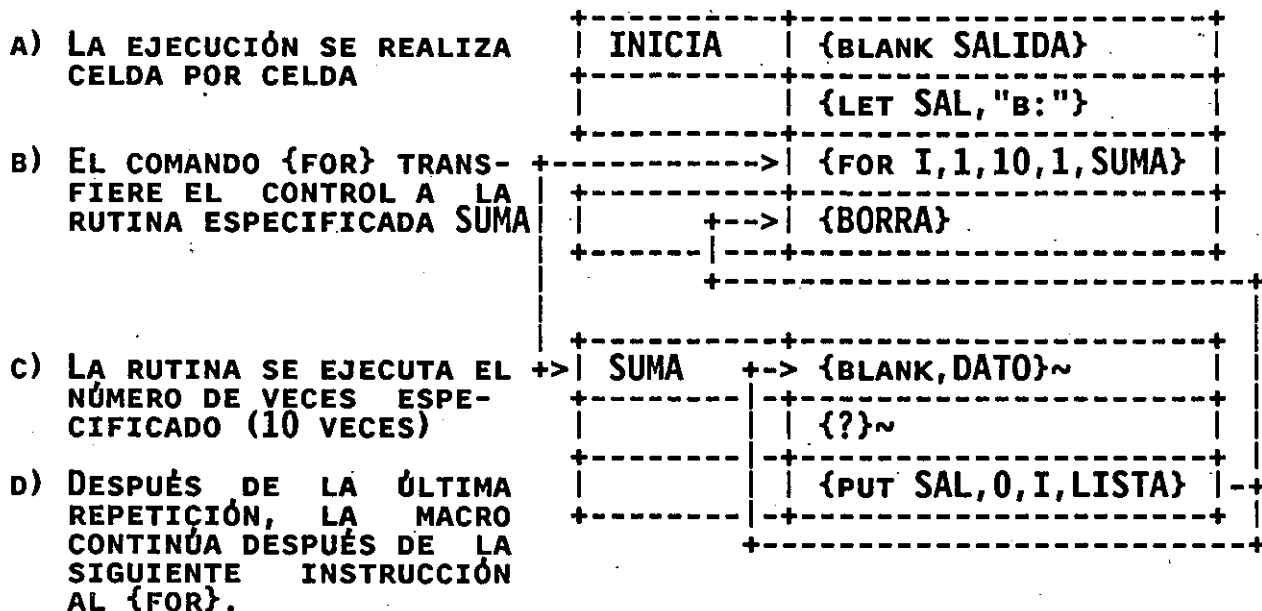
LA MANERA EXPLICITA DE TERMINAR UNA SUBROUTINA ES EL COMANDO {RETURN}. UNA SUBROUTINA TAMBIÉN TERMINA CUANDO SE ENCUENTRA UNA CELDA VACÍA, UNA CELDA NUMÉRICA O UNA CELDA CUYO VALOR SEA UN STRING VACÍO. EL COMANDO {RETURN} NO TIENE SENTIDO SI SE USA FUERA DE UNA SUBROUTINA.

REPITIENDO UNA SUBROUTINA.-

MUCHAS APLICACIONES A MENUDO INCLUYEN PROCEDIMIENTOS QUE SON EJECUTADOS REPETIDAMENTE, DURANTE UN NÚMERO DETERMINADO DE OCASIONES O HASTA QUE SE CUMPLA UNA CONDICIÓN. POR EJEMPLO, UN PROCESO QUE ACTUALIZA LOS 50 PRIMEROS REGISTROS DE UNA NÓMINA O UN PROCESO QUE ACTUALIZA TODOS LOS EMPLEADOS DE UN DETERMINADO DEPARTAMENTO.

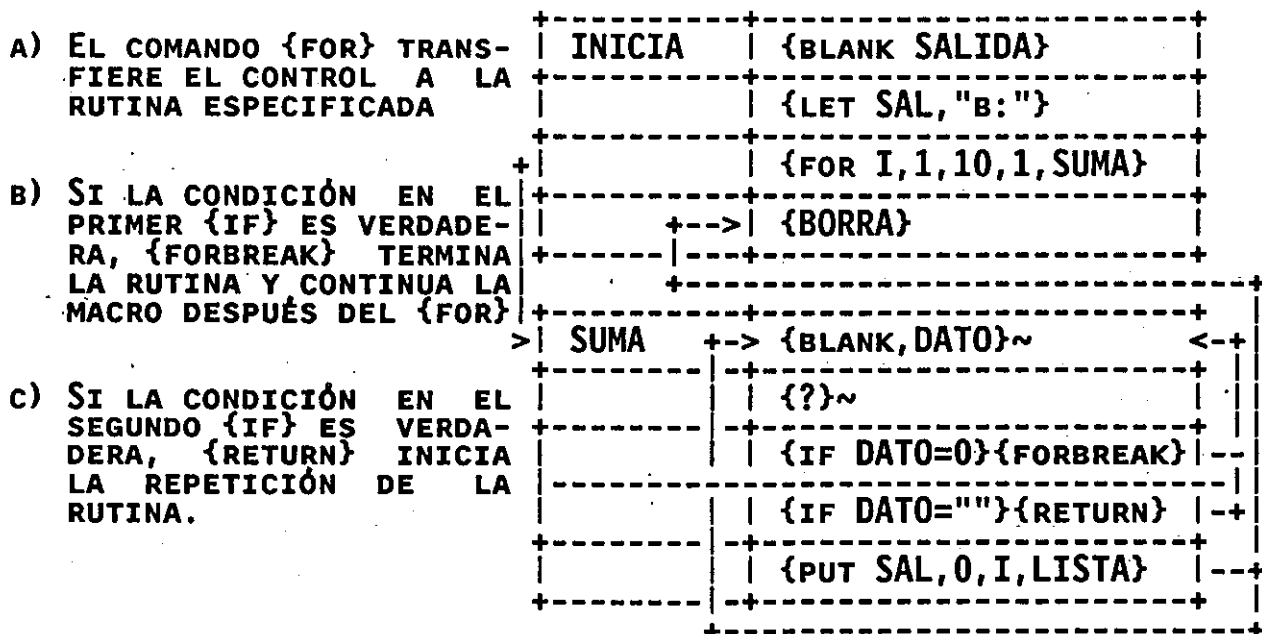
REALMENTE ES FÁCIL REALIZAR UNA MACRO QUE EJECUTE UN NÚMERO ESPECÍFICO DE OCASIONES UN PROCESO -CONSTRUYENDO EL PROCESO COMO UNA SUBROUTINA-. ESTO SE REALIZA POR MEDIO DEL COMANDO {FOR}, QUE ESPECIFICA EL NÚMERO DE REPETICIONES. DESPUÉS DE LA ÚLTIMA REPETICIÓN, EL CONTROL REGRESA A LA INSTRUCCIÓN QUE SIGUE AL COMANDO {FOR}.

NÓTESE QUE UNA SUBROUTINA QUE SE PROCESA REPETIDAMENTE PUEDE SER IMPLEMENTADA POR MEDIO DE LLAMADAS NORMALES A SUBROUTINAS Y LOS COMANDOS {IF} Y {BRANCH}, SIN EMBARGO POR MEDIO DEL COMANDO {FOR}, SE LOGRA UNA MAYOR CLARIDAD EN EL CÓDIGO Y UNA MAYOR EFICIENCIA.



EJECUCIÓN REPETIDA DE UNA SUBRUTINA CON {FOR}

EN ALGUNAS OCASIONES ES NECESARIO INTERRUPTIR LA EJECUCIÓN DE UNA SUBRUTINA INVOCADA CON {FOR}, YA SEA PARA EMPEZAR LA SIGUIENTE REPETICIÓN O TERMINAR TOTALMENTE EL {FOR}. EXISTEN DOS COMANDOS QUE PERMITEN SUSPENDER UNA REPETICIÓN (LOOP). EL COMANDO {RETURN} TERMINA LA REPETICIÓN REGRESANDO EL PROCESO A LA PRIMERA INSTRUCCIÓN DEL LOOP. EL COMANDO {FORBREAK} PERMITE INTERRUPTIR EL {FOR}, MANDANDO EL CONTROL A LA SIGUIENTE INSTRUCCIÓN AL {FOR}.



COMO SE OBSERVA EL COMANDO {FOR} EJECUTA UNA SUBROUTINA UN NÚMERO DETERMINADO DE VECES, NINGÚN COMANDO DE 1-2-3 REPITE UNA SUBROUTINA HASTA QUE UNA DETERMINADA CONDICIÓN ES SATISFECHA. ESTE PROCESO SE REALIZA CON LA IMPLEMENTACIÓN DE {BRANCH} Y LLAMADAS A SUBROUTINAS PARA CREAR LA CONSTRUCCIÓN DO-WHILE.

LA SECUENCIA QUE SIGUE EL COMANDO {FOR} ES:

{FOR CONTADOR, EMPIEZA, TERMINA, SALTA, SUBROUTINA}

- A) REEMPLAZA EL CONTADOR POR EL VALOR EMPIEZA
- B) COMPARA EL VALOR DE CONTADOR CON EL VALOR DE TERMINA. SI EL VALOR DEL CONTADOR ES MENOR O IGUAL AL VALOR DE TERMINA, SE EJECUTA LA SUBROUTINA. SI EL CONTADOR ES MAYOR QUE EL VALOR DE TERMINA YA NO SE LLAMA A LA SUBROUTINA (EL COMANDO {FOR} TERMINA)
- C) AGREGA EL VALOR DE SALTA AL VALOR DE CONTADOR
- D) REGRESA AL PASO B)

LOS ARGUMENTOS EMPIEZA, TERMINA Y SALTA PUEDEN SER UN NÚMERO, UNA FÓRMULA O EL NOMBRE DE UNA CELDA CON VALOR NUMÉRICO, FRACCIONES Y VALORES NEGATIVOS.

PASO DE VALORES A UNA SUBROUTINA. -

EN UNA LLAMADA A UNA SUBROUTINA SE PUEDEN ESPECIFICAR UNO O MAS VALORES QUE LA SUBROUTINA PUEDA PROCESAR. ESTA CAPACIDAD DE PASO DE ARGUMENTOS PERMITE A LA MISMA SUBROUTINA PROCESAR DIFERENTES DATOS EN TIEMPOS DIFERENTES.

ANTES DE QUE UN ARGUMENTO SEA PASADO A UNA SUBROUTINA PARA SER PROCESADO, EL VALOR DEBERÁ SER ALMACENADO EN ALGUNA CELDA DE LA HOJA DE TRABAJO. EL COMANDO {DEFINE} MANEJA EL ALMACENAMIENTO Y EVALUACIÓN DE LOS ARGUMENTOS DE UNA SUBROUTINA.

- | | |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| A) EN LA LLAMADA DE LA SUBROUTINA SE PASAN DOS ARGUMENTOS (C Y 5) | <pre> +-----+-----+ INICIA {HOME}REPORTE~ +-----+-----+ {AMPLIA C,5} +-----+-----+ </pre> |
| B) {DEFINE} ALMACENA LOS ARGUMENTOS EN LAS CELDAS ESPECIFICADAS. | <pre> +-----+-----+ AMPLIA {DEFINE COL,ANCHO} +-----+-----+ {GOTO}{COL}1~ +-----+-----+ </pre> |
| C) LAS INSTRUCCIONES DENTRO DE LA SUBROUTINA TOMAN LOS VALORES PASADOS A LAS CELDAS ESPECIFICADAS. | <pre> +-----+-----+ /WCS{ANCHO}~ +-----+-----+ </pre> |

COL	C
ANCHO	5

NÓTESE QUE LOS VALORES ALMACENADOS EN LAS CELDAS NO SON EXCLUSIVOS DE UNA SUBROUTINA EN PARTICULAR, ESTOS SON GLOBALES Y PUEDEN SER USADOS (O MODIFICADOS) POR UNA SUBROUTINA O POR LA MACRO PRINCIPAL.

NÓTESE TAMBIÉN QUE LOS VALORES RECIBIDOS EN LA SUBROUTINA SON ALMACENADOS COMO ETIQUETA (ALINEADOS AL LADO IZQUIERDO). SI SE REQUIERE PASAR UN ARGUMENTO COMO VALOR NUMÉRICO SE PUEDE ESPECIFICAR DE LA SIGUIENTE FORMA:

{DEFINE} ALMACENA LOS ARGUMENTOS EN LAS CELDAS ESPECIFICADAS

AMPLIA	{DEFINE COL, ANCHO:v}
	{GOTO}{COL}1~
	/WCS{ANCHO}~

LOS VALORES RECIBIDOS EN LA SUBROUTINA SE ALMACENAN COMO ETIQUETA Y COMO VALOR NUMÉRICO

COL	C
ANCHO	5

CUANDO SE PASAN ARGUMENTOS A UNA SUBROUTINA, LA PRIMERA INSTRUCCIÓN DE LA SUBROUTINA DEBE SER EL COMANDO {DEFINE}.

ANIDAMIENTO DE SUBROUTINAS.-

AL HECHO DE TENER UNA SUBROUTINA QUE LLAME A OTRA SE LE CONOCE COMO ANIDAMIENTO. EN EL LENGUAJE DE 1-2-3 SE PUEDEN TENER HASTA 31 SUBROUTINAS ANIDADAS.

CADA VEZ QUE UNA SUBROUTINA ES LLAMADA POR OTRA, LA INFORMACIÓN VA QUEDANDO ALMACENADA INTERNAMENTE EN 1-2-3 EN UN CONTADOR LLAMADO STACK DE SUBROUTINAS. SI SE REQUIERE BORRAR LA INFORMACIÓN QUE CONTIENE ESTE STACK, YA SEA PARA REGRESAR A LA MACRO PRINCIPAL O PARA TERMINAR EL PROGRAMA, SE UTILIZA EL COMANDO {RESTART}.

SIEMPRE QUE UNA MACRO ESTÁ UTILIZANDO EL STACK DE SUBROUTINAS Y OCURRE UN ERROR, ENTONCES EL STACK ES BORRADO AUTOMÁTICAMENTE. LA

EJECUCIÓN PUEDE CONTINUAR BAJO EL CONTROL DEL COMANDO {ONERROR} DE 1-2-3. EN ESTE CASO, SIN EMBARGO, 1-2-3 CONSIDERA A LA RUTINA QUE CONTIENE {ONERROR} COMO LA MACRO PRINCIPAL. CUANDO LA RUTINA TERMINA, EL CONTROL NO REGRESA A LA RUTINA PREVIA, LA MACRO SIMPLEMENTE TERMINA.

LA FORMA DE UTILIZAR EL COMANDO {ONERROR} ES:

```
+-----+
| \A          | {ONERROR RUTINA_ERROR} |
+-----+
```

```
+-----+
| RUTINA_ERROR | {LET ERROR, "HAY ERROR"} |
+-----+
```

IX MANEJO DE MENUS

EN ESTA SECCIÓN SE VERÁ COMO CONSTRUIR MENÚS QUE PERMITAN REALIZAR TODO UN PROCESO O UNA PARTE. LAS OPCIONES DE LOS MENÚS SE DESPLEGARÁN EN EL PANEL DE CONTROL, Y LA SELECCIÓN SE HACE DE LA MISMA FORMA EN QUE EL USUARIO ESTA ACOSTUMBRADO A MANEJAR LOS MENÚS DE 1-2-3.

LOS MENÚS SE VAN A INVOCAR CON LOS COMANDOS {MENUBRANCH} O {MENUCALL}. LA LISTA DE OPCIONES SE VA A DESPLEGAR EN LA SEGUNDA LÍNEA DEL PANEL DE CONTROL.

CADA OPCIÓN EN UN MENÚ ES DEFINIDA POR UNA COLUMNA. EN EL PRIMER RENGLÓN DE LA COLUMNA DEBE IR LA PALABRA QUE DESCRIBA LA OPCIÓN, SI ESTA SE ESCRIBE CON LA PRIMERA LETRA EN MAYÚSCULA EL USUARIO PODRÁ SELECCIONAR ESTA OPCIÓN CON OPRIMIR ESA LETRA. EL SEGUNDO RENGLÓN DEBE CONTENER UNA DESCRIPCIÓN QUE SE DESPLEGARÁ EN EL PANEL DE CONTROL CUANDO LA OPCIÓN ESTÉ ILUMINADA. EN EL TERCER RENGLÓN DEBERÁ IR LA PRIMERA INSTRUCCIÓN QUE DEBA REALIZARSE CUANDO ESTA OPCIÓN SEA SELECCIONADA.

```
+-----+-----+
| \A          | {MENUBRANCH MENU} |
+-----+-----+
|              | {CONTINUA}         |
+-----+-----+

```

```
-> +-----+-----+-----+
| MENU      | IMPRIMIR          | COPIAR          |
+-----+-----+-----+
|           | IMPRIME UN RANGO | COPIA UN RANGO |
+-----+-----+-----+
|           | '/PPA1..A15~G    | '/CB1..B25~C1~ |
+-----+-----+-----+

```

CON EL COMANDO {MENUBRANCH} SE INVOKA AL MENÚ PERO CUANDO TERMINA LA EJECUCIÓN DE LA RUTINA DE LA OPCIÓN ELEGIDA, EL CONTROL NO REGRESA A LA SIGUIENTE INSTRUCCIÓN DEL {MENUBRANCH}.

CON EL COMANDO {MENUCALL} EL CONTROL REGRESA A LA SIGUIENTE INSTRUCCIÓN DESPUÉS DE LA LLAMADA AL MENÚ.

X CONTROLANDO ERRORES Y PROTEGIENDO APLICACIONES

EL USO DE MENÚS REDUCE LAS POSIBILIDADES DE QUE EL USUARIO INTRODUZCA UNA INFORMACIÓN INCORRECTA EN LAS MACROS INTERACTIVAS, SIN EMBARGO SI UN ERROR OCURRE, LA EJECUCIÓN DE LA MACRO SE DETIENE Y EL CONTROL REGRESA AL USUARIO, Y NO SE PODRÍA CONTINUAR EL PROCESO PORQUE NO SE SABE EN QUE ETAPA QUEDÓ.

EL COMANDO {ONERROR} DA LA CAPACIDAD DE CONTROLAR ALGUNOS ERRORES.

OTRO ERROR QUE PODRÍA OCURRIR DURANTE LA EJECUCIÓN DE LA MACRO SERÍA QUE SE OPRIMIERA LA TECLA BREAK ACCIDENTALMENTE, EN ESTE CASO SE PUEDE ACTIVAR O DESACTIVAR ESTA TECLA CON LOS COMANDOS {BREAKON} Y {BREAKOFF}.

EL COMANDO {BREAKOFF} DESHABILITA LA TECLA BREAK HASTA QUE SE VUELVA A HABILITAR CON EL COMANDO {BREAKON}.

\k	{BREAKOFF}
	{ONERROR RECUPERA}
	{RUTINA1}
	{RUTINA2}
	{BREAKON}

RECUPERA	'/QY
----------	------

EN LA EJECUCIÓN DE ESTA MACRO SE DESHABILITA LA TECLA BREAK, Y EN CASO DE QUE OCURRA UN ERROR EN ALGUNA DE LAS SUBRUTINAS EL CONTROL PASARÁ A LA RUTINA RECUPERA. EN ESTA RUTINA PODRÍA CONTROLARSE LA SALIDA DE LA HOJA SIN QUE LA INFORMACIÓN SEA SALVADA.


```

+-----+
| \E    | {OPEN PERSONAL, "W"} |
+-----+
|       | {WRITELN LINEA}      |
+-----+
|       | {CLOSE}              |
+-----+

```

```

+-----+
| LINEA | PEDRO LOPEZ |
+-----+

```

```

+-----+
| P | E | D | I | R | I | O | | L | I | O | P | E | Z | I | |
+-----+

```

<----- ARCHIVO PERSONAL

NAVEGANDO DENTRO DE UN ARCHIVO. -

```

+-----+
| POSICION | 11 |
+-----+
| COLOCA   | 4  |
+-----+
| TOTAL    | 20 |
+-----+

```

```

PRIMER CARACTER EN LA POSICIÓN 0
      |
      v
+-----+
| A | 4 | 9 | 7 | B | 0 | 6 | 4 | C | 9 | 9 | 9 | D | 1 | 0 | 0 | E | 2 | 1 | 5 |
+-----+
      ^
      |
      {SETPOS COLOCA}
      REPOSICIONA EL APUNTADOR DEL CARACTER

      {GETPOS POSICION}
      MUESTRA EN QUE POSICIÓN SE ENCUENTRA EL APUNTADOR

      - ÚLTIMO CARACTER POSICIÓN
      19

```

```

+-----+
| A | 4 | 9 | 7 | B | 0 | 6 | 4 | C | 9 | 9 | 9 | D | 1 | 0 | 0 | E | 2 | 1 | 5 |
+-----+

```

```

{SETPOS COLOCA}
REPOSICIONA EL
APUNTADOR DEL
CARACTER

```

```

{FILESIZE TOTAL}
MUESTRA EL TAMANO
DEL ARCHIVO

```

RESUMEN DE COMANDOS

{LET} GUARDA UN NÚMERO O UNA ETIQUETA EN ALGUNA CELDA ESPECIFICADA.

SINTAXIS:

{LET LOCALIZACIÓN, VALOR-NUMÉRICO}

{LET LOCALIZACIÓN, ETIQUETA}

EJEMPLO:

{LET VALOR, 495}

{LET CADENA, "LÓPEZ"}

{BLANK} BORRA EL CONTENIDO DE UNA CELDA O DE CADA CELDA DE ALGÚN RANGO ESPECIFICADO.

SINTAXIS:

{BLANK LOCALIZACIÓN}

EJEMPLO:

{BLANK TOTAL}

{BLANK TABLA}

{CALC} RECALCULA TODAS LAS FÓRMULAS EN LA HOJA DE TRABAJO.

SINTAXIS:

{CALC}

{RECALC} RECALCULA LAS FÓRMULA EN EL RANGO ESPECIFICADO, RENGLÓN POR RENGLÓN.

SINTAXIS:

{RECALC LOCALIZACIÓN}

EJEMPLO:

{LET GANANCIA, PRECIO-COSTO}
{RECALC}

{BRANCH} CAMBIA LA SECUENCIA DE EJECUCIÓN DEL PROGRAMA A LA CELDA ESPECIFICADA.

SINTAXIS:

{BRANCH LOCALIZACIÓN}

EJEMPLO:

{BRANCH CALCULA}

{DISPATCH} EJECUTA EL COMANDO **{BRANCH}** INDIRECTAMENTE. LA EJECUCIÓN DE LA MACRO CONTINÚA EN LA RUTINA ESPECIFICADA.

EJEMPLO:

**{LET NOMBRE_RUTINA,"INICIO"}
{DISPATCH NOMBRE_RUTINA}**

{QUIT} TERMINA LA EJECUCIÓN DE LA MACRO.

SINTAXIS:

{QUIT}

EJEMPLO:

**{IF BANDERA="NO"}{QUIT}
{BRANCH CONTINUA}**

LLAMADO DE SUBRUTINAS. LLAMADA A UNA SUBRUTINA OCASIONA QUE LA SECUENCIA DE DE EJECUCIÓN DEL PROGRAMA CONTINÚE EN OTRA LOCALIZACIÓN Y CUANDO TERMINA REGRESA A EJECUTARSE LA INSTRUCCIÓN SIGUIENTE A LA LLAMADA DE LA SUBRUTINA.

SINTAXIS:

{NOMBRE-SUBRUTINA}

EJEMPLO:

**{IF BALANCE=1000}{AJUSTA}
{CONTINUA}**

{DEFINE} EVALÚA Y ALMACENA CADA ARGUMENTO PASADO A LA SUBROUTINA.

SINTAXIS:

{DEFINE CELDA1, CELDA2, ...}

EJEMPLO:

{PROCESO 500.25, TASA}

{DEFINE CELDA1, CELDA2}

{FOR} EJECUTA UNA SUBROUTINA UN CIERTO NÚMERO DE VECES.

SINTAXIS:

{FOR CONTADOR, EMPIEZA, TERMINA, SALTA, NOMBRE-SUBROUTINA}

EJEMPLO:

{FOR I, 1, 5, 1, PROCESO}

{FOR CONTADOR, 10, 0, -2, RUTINA}

{RETURN} TERMINA UNA SUBROUTINA. EL CONTROL REGRESA JUSTO AL PUNTO DESPUÉS DE LA LLAMADA DE LA SUBROUTINA.

SINTAXIS:

{RETURN}

EJEMPLO:

{IF BANDERA="CONTINUA"}{BRANCH MAS}
{RETURN}

{FORBREAK} TERMINA UNA SUBROUTINA LLAMADA POR **{FOR}**

SINTAXIS:

{FORBREAK}

EJEMPLO:

{IF MAXIMO>20000}{FORBREAK}
{LET MAXIMO, MAXIMO+30}

{RESTART} LIMPIA LA PILA DE LA SUBROUTINA.

SINTAXIS:

{RESTART}

EJEMPLO:

{IF RESPUESTA="NO"}{RESTART}{BRANCH EMPIEZA}

{?} PERMITE AL USUARIO USAR UN NÚMERO ILIMITADO DE TECLAS DURANTE LA EJECUCIÓN DE UNA MACRO.

SINTAXIS:

{?}

EJEMPLO:

{GOTO}{?}~

**{GOTO}DESCUENTO~
{?}/100~**

{GETLABEL} SUSPENDE LA EJECUCIÓN DE LA MACRO HASTA QUE EL USUARIO TECLEE ALGUNA CADENA. CUANDO EL USUARIO OPRIME RETURN LA CADENA ES ALMACENADA Y LA EJECUCIÓN DE LA MACRO CONTINÚA.

SINTAXIS:

{GETLABEL "PREGUNTA", LOCALIZACIÓN}

EJEMPLO:

{GETLABEL "TECLEE SU NOMBRE, POR FAVOR: ", NOMBRE}

{GETNUMBER} SUSPENDE LA EJECUCIÓN DE LA MACRO HASTA QUE EL USUARIO TECLEE UN NÚMERO. CUANDO EL USUARIO PRESIONA LA TECLA RETURN, EL NÚMERO ES ALMACENADO Y LA EJECUCIÓN DE LA MACRO CONTINÚA.

SINTAXIS:

{GETNUMBER "PREGUNTA-NÚMERO", LOCALIZACIÓN}

EJEMPLO:

{GETNUMBER "PAGO TOTAL: ", MONTO}

{GET} SUSPENDE LA EJECUCIÓN DE LA MACRO HASTA QUE EL USUARIO OPRIMA UNA TECLA.

SINTAXIS:

{GET LOCALIZACIÓN}

EJEMPLO:

{GET LLAVE}

{LOOK} CREA UNA ETIQUETA QUE CONSISTE DEL PRIMER CARACTER QUE ESTÁ ALMACENADO EN LA CELDA.

SINTAXIS:

{LOOK LOCALIZACIÓN}

EJEMPLO:

{LOOK RESPUESTA}

{WAIT} SUSPENDE TODO EL PROCESO POR EL TIEMPO ESPECIFICADO.

SINTAXIS:

{WAIT NÚMERO-SERIAL}

EJEMPLO:

{WAIT @NOW+@TIME(0,0,5)}

{MENUBRANCH} DESPLIEGA UN MENÚ EN EL PANEL DE CONTROL Y LLAMA A LA RUTINA ASOCIADA CON LA SELECCIÓN HECHA POR EL USUARIO.

SINTAXIS:

{MENUBRANCH LOCALIZACIÓN}

EJEMPLO:

{MENUBRANCH PRIMERO}

{MENUCALL} DESPLIEGA UN MENÚ EN EL PANEL DE CONTROL Y LLAMA A LA SUBROUTINA ASOCIADA CON LA SELECCIÓN HECHA POR EL USUARIO Y LA EJECUCIÓN DE LA MACRO CONTINÚA EN LA INSTRUCCIÓN SIGUIENTE A LA LLAMADA DEL MENÚ.

SINTAXIS:

{MENUCALL LOCALIZACIÓN}

EJEMPLO:

{MENUCALL PRIMERO}

{WINDOWSOFF} CONGELA EL ÁREA DE DESPLEGADO DE LA PANTALLA.

SINTAXIS:

{WINDOWSOFF}

EJEMPLO:

{WINDOWSOFF}
{GOTO}AREA~

{WINDOWSON} DESCONGELA EL ÁREA DE DESPLEGADO DE LA PANTALLA. ES EL COMANDO CONTRARIO DE **{WINDOWSOFF}**.

SINTAXIS:

{WINDOWSON}

EJEMPLO:

{WINDOWSOFF}
{DOWN 8}
{WINDOWSON}

{PANELOFF} CONGELA EL DESPLEGADO DEL PANEL DE CONTROL, INCLUYENDO EL INDICADOR DE MODO.

SINTAXIS:

{PANELOFF}

EJEMPLO:

{WINDOWSOFF}{PANELOFF}
/CENTRADA~SALIDA~

{PANELON} DESCONGELA EL DESPLEGADO EN EL PANEL DE CONTROL.
ES EL COMANDO CONTRARIO DE {PANELOFF}.

SINTAXIS:

{PANELON}

EJEMPLO:

{WINDOWSOFF}{PANELOFF}
'/ENTRADA~SALIDA~
{PANELON}{WINDOWSON}

{INDICATE} ESTE COMANDO TIENE DOS FORMAS:-
 {INDICATE CADENA} E **{INDICATE}**.
 LA PRIMERA FORMA DESPLIEGA LA CADENA ESPECIFICADA
 EN EL INDICADOR DE MODO. LA SEGUNDA FORMA REGRESA
 EL CONTROL DEL INDICADOR DE MODO AL 1-2-3.

SINTAXIS:

{INDICATE CADENA}

{INDICATE}

EJEMPLO:

{INDICATE HOLA}

{INDICATE " 1 "}

{BEEP} EMITE UN SONIDO

SINTAXIS:

{BEEP}

{BEEP NÚMERO}

EJEMPLO:

{BEEP 0}

{BEEP 3}

{ONERROR} ESTABLECE UN PROCEDIMIENTO A SEGUIR SI EN 1-2-3 OCURRE UN ERROR DURANTE LA EJECUCIÓN DE LA MACRO.

SINTAXIS:

{ONERROR LOCALIZACIÓN}

{ONERROR LOCALIZACIÓN, LOCALIZACIÓN-MENSAJE}

EJEMPLO:

{ONERROR RECUPERA}

{BREAKOFF} DESHABILITA LA TECLA BREAK DURANTE LA EJECUCIÓN DE LA MACRO.

SINTAXIS:

{BREAKOFF}

{BREAKON} HABILITA LA TECLA BREAK. ES EL COMANDO CONTRARIO A {BREAKOFF}.

SINTAXIS:

{BREAKON}

EJEMPLO:

{BREAKOFF}
{GETLABEL "DEME SU NOMBRE: ", NOMBRE}
{BREAKON}

{OPEN} HACE ACCESIBLE UN ARCHIVO PARA PROCESARLO CARACTER POR CARACTER O LÍNEA POR LÍNEA.

SINTAXIS:

{OPEN ARCHIVO, MODO}

MODO	R	LECTURA
	W	ESCRITURA
	M	MODIFICACIÓN

EJEMPLO:

{OPEN LISTA, "R"}

{CLOSE} TERMINA EL ACCESO AL ARCHIVO QUE SE ENCUENTRA ABIERTO.

SINTAXIS:

{CLOSE}

EJEMPLO:

**{OPEN NOMINA, "R"}
{CLOSE}**

{READ} TRANSFIERE UN NÚMERO ESPECIFICADO DE BYTES DE UN ARCHIVO A LA HOJA DE TRABAJO.

SINTAXIS:

{READ CONTADOR, CELDA}

EJEMPLO:

{READ 8, ALMACENA}

{READLN} TRANSFIERE UNA LÍNEA DE UN ARCHIVO TEXTO A LA HOJA DE TRABAJO.

SINTAXIS:

{READLN CELDA}

EJEMPLO:

{READLN ALMACENA}

{WRITE} SACA UNA CADENA DE CARACTERES A UN ARCHIVO.

SINTAXIS:

{WRITE CADENA}

EJEMPLO:

{WRITE "UNO"}

{WRITELN} SACA UNA CADENA DE CARACTERES A UN ARCHIVO. (UNA LÍNEA)

SINTAXIS:

{WRITELN CADENA}

EJEMPLO:

{WRITELN "LINEA DE PRUEBA"}

{FILESIZE} PROPORCIONA EL TAMANO DEL ARCHIVO QUE SE ENCUENTRA ABIERTO.

SINTAXIS:

{FILESIZE CELDA}

EJEMPLO:

{OPEN "PRIMERO" "R"}
{FILESIZE LONGITUD}

{SETPOS} CAMBIA LA POSICIÓN DEL APUNTAOR DE BYTES EN EL ARCHIVO QUE SE ENCUENTRA ABIERTO.

SINTAXIS:

{SETPOS POSICIÓN}

EJEMPLO:

{FILESIZE TOTAL}
{SETPOS TOTAL}

{GETPOS} PROPORCIONA EL NÚMERO DE POSICIÓN EN LA QUE SE ENCUENTRA EL APUNTAOR DEL ARCHIVO ABIERTO.

SINTAXIS:

{GETPOS POSICIÓN}

EJEMPLO:

{GETPOS LUGAR}

RESUMEN DE FUNCIONES

FUNCIONES MATEMÁTICAS:

- @ABS(X) VALOR ABSOLUTO DE X
@ROUND(X,N) REDONDEA UN NÚMERO A N DECIMALES

FUNCIONES LÓGICAS:

- @FALSE ENTREGA EL VALOR 0 (FALSO)
@IF(COND,X,Y) ENTREGA EL VALOR "X" SI LA CONDICIÓN ES VERDADERA Y EL VALOR "Y" SI LA CONDICIÓN ES FALSA
@THRU ENTREGA EL VALOR 1 (VERDADERO)
@@(DIR.CELDA) ENTREGA EL VALOR DE LA CELDA REFERENCIADA POR LA DIRECCIÓN DE LA CELDA DIR.CELDA
@CHOOSE(X,V0,V1,...VN) DE ACUERDO AL VALOR DE X, TOMA EL VALOR DEL ELEMENTO DE LA LISTA V
@COLS(RANGO) REGRESA EL NÚMERO DE COLUMNAS QUE EXISTE EN EL RANGO.
@HLOOKUP(X,RANGO,R) BUSCA EN UN RANGO HORIZONTALMENTE EL VALOR X EN EL RENGLÓN R Y REGRESA EL VALOR DE ESA CELDA
@INDEX(RANGO,COLUMNA,RENGLÓN) REGRESA EL VALOR DE LA CELDA QUE INTERSECTA LA COLUMNA Y EL RENGLÓN
@ROWS(RANGO) REGRESA EL NÚMERO DE RENGLONES QUE EXISTE EN EL RANGO
@VLOOKUP(X,RANGO,C) BUSCA EN UN RANGO VERTICALMENTE EL VALOR X EN LA COLUMNA C, Y REGRESA EL VALOR DE ESA CELDA

FUNCIONES STRING:

- @EXACT(STRING1,STRING2) CHECA SI LOS DOS STRING SON IGUALES, REGRESA 1 EN CASO AFIRMATIVO Y 0 DE LO CONTRARIO
@FIND(STRING-BUSCADO,STRING,NUMERO-INICIO) REGRESA LA POSICIÓN DENTRO DEL STRING DONDE SE CUMPLE LA PRIMERA OCURRENCIA DEL STRING BUSCADO

@LEFT (STRING, N) REGRESA LOS PRIMEROS N CARACTERES DE LA IZQUIERDA DEL STRING

@LENGTH (STRING) REGRESA EL NÚMERO DE CARACTERES QUE CONTIENE EL STRING

@LOWER (STRING) CONVIERTE TODAS LAS LETRAS DEL STRING A MINÚSCULAS

@MID (STRING, NÚMERO-INICIO, N) EXTRAER N CARACTERES DEL STRING A PARTIR DEL NÚMERO DE INICIO

@N (RANGO) REGRESA EL VALOR DE LA ESQUINA SUPERIOR IZQUIERDA DEL RANGO CONVERTIDO EN NÚMERO

@PROPER (STRING) CONVIERTE TODAS LAS PRIMERAS LETRAS DE LAS PALABRAS DEL STRING A MAYÚSCULAS

@REPEAT (STRING, N) REPITE EL STRING N-VECES

@REPLACE (STRING-ORIGINAL, NÚMERO-INICIAL, N, STRING-NUEVO) QUITA N CARACTERES DEL STRING ORIGINAL, A PARTIR DEL NÚMERO DE INICIO E INSERTA EL STRING NUEVO.

@RIGHT (STRING, N) REGRESA LOS ÚLTIMOS N CARACTERES DEL DEL STRING

@S (RANGO) REGRESA EL VALOR DE LA ESQUINA SUPERIOR DEL RANGO CONVIRTIÉNDOLO EN STRING

@STRING (X, N) CONVIERTE EL NÚMERO X A STRING CON N DECIMALES

@TRIM (STRING) SUPRIME ESPACIOS EN BLANCO ENTRE LAS PALABRAS DEL STRING

@UPPER (STRING) CONVIERTE TODAS LAS LETRAS DEL STRING A MAYÚSCULAS

@VALUE (STRING) CONVIERTE EL STRING EN EL VALOR NUMÉRICO QUE REPRESENTA

FUNCIONES DE FECHA Y HORA:

@DATEVALUE (STRING-FECHA) REGRESA EL NÚMERO SERIAL DE LA FECHA DADA EN EL STRING-FECHA

@DAY (FECHA) REGRESA EL NUMERO DE DIA

@HOUR (FECHA) REGRESA EL NÚMERO DE HORA

@MINUTE(FECHA) REGRESA EL NUMERO DE MINUTO EN LA HORA
@MONTH(FECHA) REGRESA EL NÚMERO DE MES
@NOW REGRESA LA FECHA Y HORA DEL SISTEMA
@YEAR(FECHA) REGRESA EL NÚMERO DE HORA

FUNCIONES ESTADÍSTICAS:

@COUNT(LISTA) CUENTA EL NÚMERO DE ELEMENTOS DE UNA LISTA
@SUM(LISTA) SUMA LOS VALORES DE TODOS LOS ELEMENTOS DE UNA LISTA
@AVG(LISTA) OBTIENE EL VALOR PROMEDIO DE LOS ELEMENTOS DE UNA LISTA