



manual de usuario en español

**Copyright © (1990-2022) SW-Tools ApS**  
**Duevej 23**  
**DK-2680 Solrød Strand**  
**Denmark**  
**Phone: +45) 33 33 05 56**  
**Mail: swtools@swtools.com**  
**www: www.swtools.com**

## **Calculos y subfunciones**

**22/11/01 / 2022-09-01 008.384**

## Contenidos

Contenidos .....	2
1. Introduccion.....	6
1.1. Ejemplos.....	7
1.1.1. <u>IF..ELSE</u> - Condicional estados.....	8
1.1.2. <u>BEGIN..END</u> - Estados del bloque.....	9
1.1.3. <u>START/END...NEXT...REPEAT</u> - Lazos.....	10
1.1.4. <u>NO, Y, O</u> - Operaciones logicas .....	11
1.1.5. <u>REM, /*</u> - Comentarios.....	12
1.1.6. <u>GOTO</u> Saltar etiqueta.....	13
1.1.6.1. <u>ON...GOTO/GOSUB</u> - Condicional saltar / subrutinallamada .....	14
1.1.7. <u>GOSUB</u> Llamando subrutinas .....	15
1.1.7.1. <u>RETURN</u> Volviendo de una subrutina .....	16
1.2. Campos .....	17
1.2.1. <u>#xx o kk#xx</u> - Campos de una fila .....	18
1.2.1.1. <u>#xx(de,a)</u> - Parte de los campos .....	19
1.2.1.2. <u>#xx(no)</u> - Etiquetas de los campos .....	20
1.2.1.3. Conversion entre campos numericos y textos.....	21
1.2.2. <u>SY#xx</u> - Sistema de los campos .....	22
1.2.2.1. <u>#DD, #PD</u> - Fecha de hoy como la fecha .....	23
1.2.2.2. <u>#PP</u> - Numero de pagina .....	24
1.2.2.3. <u>#SN</u> - Nombre del sistema .....	25
1.2.2.4. <u>#OK</u> - Resultados despues de estar leyendo de una fila.....	26
1.2.2.5. <u>#UN</u> Nombre del usuario .....	27
1.2.2.6. <u>#LIN</u> numero de la linea y <u>#LOF</u> lineas en forma.....	28
1.2.2.7. <u>#LEVEL</u> - Actual nivel total .....	29
1.2.2.8. <u>kk#RECNO</u> - El pasado numero de registro usado de la fila kk .....	30
1.2.3. <u>WW#xx</u> - Campos libre (Campos de trabajo) .....	31
1.2.3.1. <u>#Dntext</u> - Entrar fecha.....	32
1.2.3.2. <u>#Ptexto</u> - Cuadros de los campos .....	33
2. Funciones aritmeticas.....	34
2.1. <u>ABS</u> - El valor absoluto de un numero.....	35
2.2. <u>FNH</u> - Redondear numero - no decimales .....	36
2.3. <u>FNR</u> - Redondear numero a 2 decimales.....	37
2.4. <u>FRA</u> - Calcular la fraccion de un numero.....	38
2.5. <u>INT</u> - El valor integro de un numero .....	39
2.6. <u>NO</u> - Negacion logica .....	40
2.7. <u>POW</u> - Razon no poder.....	41
2.8. <u>RUN</u> - Redondeando a x decimales .....	42
2.9. <u>RUND</u> - Definicion del FNR redondeando la funcion.....	43
2.10. <u>SGN</u> - Parar si el numero es negativo, zero o positivo .....	44
2.11. <u>SQR</u> - Calcular el cuadrado de la raiz de un numero .....	45
3. Funciones de la cuerda .....	46
3.1. <u>CONV</u> - Cambiar caracteres en un texto.....	47
3.2. <u>EDIT</u> - Editando de una figura integra.....	48
3.3. <u>FIND</u> - Fin del texto en el texto del campo .....	49
3.4. <u>LEN</u> - Longitud de un texto.....	50
3.5. <u>LOWER</u> - Convertir el texto a mas bajo que la caja de caracteres .....	51
3.6. <u>NAME</u> - Extraccion de Christian y nombre pasado.....	52
3.7. <u>NUMBER</u> - Conversion de 'sucio' numeros .....	53
3.8. <u>NUMS</u> - Conversion del numero de texto del campo .....	54
3.9. <u>PACK</u> - Empaquetacion de un numero.....	55
3.10. <u>SMAA</u> - Convertir texto a mas alto/mas bajo de la caja - nombres.....	56

3.11. <u>SOGE</u> - Creacion de la busqueda de la llave de una direccion del campo .....	57
3.12. <u>SPOFF</u> - Remover delante y detras de los espacios en el texto .....	58
3.13. <u>UNPACK</u> - Desempaquetacion de un numero.....	59
3.14. <u>UPPER</u> - Convertir el texto en el mas alto de la caja .....	60
3.15. <u>USING</u> - Editando un numero.....	61
4. Dígito parado y valor.....	62
4.1. <u>CCODE</u> - Texto del campo parado (DATAMASTER codigo del texto parado).....	63
4.2. <u>CODE</u> - OCR parada .....	64
4.3. <u>CHEX</u> - Modulo 11 parado .....	65
4.4. <u>VALCH</u> - Parar si el texto encontrado en el valor del rango .....	66
4.5. <u>VALID</u> - Parar si el numero encontrado en los numeros validos .....	67
5. Manipulacion de las funciones de la fecha .....	68
5.1. <u>DATE</u> - Fecha actual YYYYMMDD .....	69
5.2. <u>DATECALC</u> - Calcular una fecha .....	70
5.3. <u>DIA</u> - Descripcion de una fecha - textoforma .....	71
5.4. <u>FNA</u> - Convertir fecha a un numero de dias de año 0 .....	72
5.5. <u>FNB</u> - Convertir numero de dias de año 0 a fecha.....	73
5.6. <u>FND</u> - Fecha conversion .....	74
5.7. <u>FNE</u> - Convertir la fecha a un numero del mes .....	75
5.8. <u>FNF</u> - Convertir fecha a numero del dia, 360 dias/año.....	76
5.9. <u>FNO</u> - Convertir fecha a DDMMYY .....	77
5.10. <u>FNU</u> - Convertir fecha a dia de la semana.....	78
5.11. <u>FNV</u> - Convertir fecha a no de la semana a fecha .....	79
5.12. <u>FNY</u> - Convertir fecha a YYYYMMDD .....	80
5.13. <u>MES</u> - texto generado con descripcion de un mes .....	81
5.14. <u>TIEMPO</u> - Tiempo actual TTMMSS .....	82
5.15. <u>WDIA</u> - Describir el dia de la semana de la fecha.....	83
5.16. <u>SEMANA</u> - Convertir fecha a semanano o semanano a fecha .....	84
5.17. <u>WORKD</u> - Calcular el numero de dias trabajados entre las fechas .....	85
6. Mano de multiples campos.....	86
6.1. <u>LET</u> - Calculando varios campos a tiempo.....	87
6.1.1. <u>LET</u> - Asignar valores a los campos entre IQ programas (IQ) .....	88
6.1.2. <u>LET</u> - Creando nuevas filas (RAP) .....	89
6.2. <u>CLEAR</u> - tocar todos los campos en una fila (RAP) .....	90
6.3. <u>CLRFLAG</u> - Limpiar opciones para los campos en la pantalla (IQ) .....	91
6.4. <u>COLOR</u> - Set background caja de color para un numero de campos .....	92
6.5. <u>COLORF</u> - Set foreground texto color para un numero de campos .....	93
6.6. <u>DIALOG</u> - Funcion adicional para la entrada.....	94
6.7. <u>GETFLAG</u> - Poner opciones para los campos en la pantalla (IQ).....	95
6.8. <u>SETFLAG</u> - Set opciones para campos en pantalla (IQ) .....	96
6.9. <u>ZERO</u> - Zero un numero de los campos.....	97
7. control del registro.....	98
7.1. <u>CHAIN</u> - Empezar el proximo registro o otro programa(RAP) .....	99
7.1.1. <u>CHAINR</u> - Cadena del programa o comando externo directamente (RAP).....	100
7.1.2. <u>CHAIN</u> - Cadena IQ programa o comando externo (IQ) .....	101
7.2. <u>WAIT</u> - Esperar para el programa acabar (IQ).....	102
7.3. <u>COMPILAR</u> - Compilar un registro (RAP).....	103
7.4. <u>EXIT</u> - Salida del registro (RAP) .....	104
7.4.1. <u>EXIT</u> - Cerrar IQ programa o ventana (IQ) .....	105
7.5. <u>KEYS</u> - Externa empezar/parar rangos (RAP).....	106
7.6. <u>INDICE</u> - Set indice y empezar/parar valor para el registro (RAP) .....	107
7.7. <u>LTOT</u> - el nivel total mas bajo (RAP).....	108
7.8. <u>MTOT</u> - Nivel total mas alto (RAP) .....	109
7.9. <u>MESS</u> - Display mensaje en pantalla.....	110
7.10. <u>NOPAS</u> - No contraseña/nombre del usuario en este registro (RAP) .....	111

7.11. <u>PAS</u> - Set contraseña/nombreusuario (RAP).....	112
7.12. <u>PARAMS</u> - Funcion para añadir registro al empezar los parametros (RAP).....	113
7.13. <u>RETURN</u> - Volver para los calculos .....	114
7.14. <u>SORTKEY</u> - Insertando clase de llave adicional (RAP) .....	115
7.15. <u>SORTWORK</u> - Usando una especifica clase de fila (RAP) .....	116
7.16. <u>WHEN</u> - Cuando realizas los calculos (RAP).....	117
8. Marcar control .....	118
8.1. <u>COPIES</u> - Numero de copias marcadas (RAP).....	119
8.2. <u>PAGE</u> - Cambiar registro de distribucion de la pagina (RAP).....	120
8.3. <u>PRINT</u> - Marcar lineas de registros de la distribucion (RAP).....	121
8.3.1. <u>PRINT</u> - Marcar salida de control (RAP.) .....	122
8.3.2. <u>PRINT(?=</u> - Marcar las características de la pregunta (RAP.) .....	123
8.4. <u>PRINT(LAB=</u> - Funcion de la etiqueta (RAP).....	124
8.5. <u>PRINTER</u> - Marcar seleccion (RAP.) .....	125
8.5.1. <u>PRINTER</u> - Multiples marcas en la salida (RAP).....	126
8.6. <u>PRTTOTAL</u> - Manual de control de la marca de fuera (RAP).....	127
8.7. <u>SCRPR</u> - Rellamar pantalla marcada (IQ).....	128
9. Leyendo filas.....	129
9.1. <u>READ</u> - Leer el registro de una fila .....	130
9.2. <u>READH</u> - leer registro con marca opcional del encabezado.....	131
9.3. <u>READR</u> - Leer registro usando el numero de registro .....	132
9.4. <u>READX</u> - Leer registro usando el numero de registro relativo .....	133
9.5. <u>START</u> - Set indice y rango para una fila .....	134
9.6. <u>NEXT</u> - Coger proximo registro en rango .....	135
9.7. <u>REPEAT</u> - Repetir leyendo NEXT .....	136
9.8. <u>GETKEY</u> - Coger actual valor de la llave .....	137
9.9. <u>END</u> - Set fin rango para una fila despues de START.....	138
9.10. <u>PRIOR</u> - Coger registros previos en rango.....	139
9.11. <u>VELOCIDAD</u> - Optimizar lectura estrategica .....	140
10. Escribiendo a filas .....	141
10.1. <u>UPDATE</u> - Permitir actualizar filas .....	142
10.2. <u>REWRITE</u> - Reescribir registro en la fila .....	143
10.3. <u>INSERT</u> - Insertar nuevo registro en la fila .....	144
10.4. <u>DELETE</u> - Suprimir un registro en una fila.....	145
10.5. <u>WRITE</u> - Escribir un registro en una fila .....	146
11. Exportar / Importar de las filas externas.....	147
11.1. <u>EXPORT</u> - Exportar de la fecha a un texto de la fila.....	148
11.2. <u>IMPORT</u> - Importar fecha de la fila del texto (RAP) .....	150
11.2.1. <u>IMPOCONT</u> - Continuacion del importe (RAP) .....	151
11.2.2. <u>IMPONEXT</u> - Importe del proximo registro (RAP) .....	152
11.2.3. <u>IMPOESTE</u> - Reimporte de este registro (RAP).....	153
11.3. <u>FTP</u> - Procesador Transferido de la fila .....	154
12. Multiples compañías y union de las filas .....	155
12.1. <u>ACCESS</u> - Controlar si fila existe (IQ).....	156
12.2. <u>COMNO</u> - Coger compañía actual id .....	157
12.3. <u>ENDSUM</u> - Total grande adiciona cuando estan usando mas filasprincipales .....	158
12.4. <u>FILENAME</u> - Actual nombrefila para una fila abierta .....	159
12.5. <u>OPEN</u> - Abrir una fila con un nombre especifico .....	160
12.5.1. <u>OPEN</u> - Temporamente cerrar filas.....	161
12.6. <u>MERGE</u> - Uniendo mas de dos filas principales en un registro (RAP) .....	162
12.7. <u>OPCOM</u> - Abrir filas en diferentes compañías .....	163
13. IQ/DATAMASTER funciones .....	164
13.1. <u>DISABLE</u> - Estropear la entra para un programa (IQ).....	165
13.2. <u>DISP</u> - Display de cambiar los campos (IQ) .....	166
13.3. <u>DOFUNCTION</u> - Ejecutar funcion externa (IQ) .....	167

13.4. <u>ENABLE</u> - Permitir la entrada para un programa (IQ) .....	168
13.5. <u>FOCUS</u> - Activar programa (IQ).....	169
13.6. <u>FUNC</u> - Actualizacion del modo actual para un registro (IQ) .....	170
13.7. <u>GETINFO</u> - Coger programa adicional de informacion (IQ/DM) .....	171
13.8. <u>HELP</u> - Display caja con ayuda para campo (IQ).....	172
13.9. <u>ISACTIVE</u> - Preguntar si el programa esta activado (IQ).....	173
13.10. <u>KEYON</u> - Interruptor de la llave de entrada en el campo ON/OFF (IQ) .....	174
13.11. <u>LINE</u> - Cobrar o set el numero de linea actual (IQ/DM).....	175
13.12. <u>LOOP</u> - Llamar una rutina para todos los registros en la memoria de la linea (IQ).....	176
13.13. <u>MENUCH</u> - Flip menu parado flag (IQ) .....	177
13.14. <u>MENUS</u> - Menu control (IQ) .....	178
13.15. <u>MENUUPD</u> - Añadir/Control menu (IQ) .....	179
13.16. <u>NEXTFLD</u> - Saltar a la entrada del campo (IQ).....	180
13.17. <u>NEXTFLDSEQ</u> - Saltar la entrada de un campo en la secuencia (IQ) .....	181
13.18. <u>OBJECTADDSTRING</u> - Añadir cuerda al objeto (IQ) .....	182
13.19. <u>OBJECTCLEAR</u> - Limpiar contenidos del objeto (IQ).....	183
13.20. <u>OBJECTGETSTRING</u> - Coger indice de un objeto seleccionado del articulo(IQ/DM) .....	184
13.21. <u>PLSNEXT</u> - Preparar y leer fila principal (IQ) .....	185
13.22. <u>SEQ</u> - Cambiar la entrada de la secuencia (IQ).....	186
13.23. <u>SETUPD</u> - Marca una fila en una linea para actualizacion (IQ).....	187
13.24. <u>SHOW</u> - Permitir/estropear/Demostrar/Esconder un campo (IQ/DM).....	188
13.25. <u>SUPER</u> - Preparar superindice buscado (IQ) .....	189
13.26. <u>TRANSMIT</u> - Actualizar otro IQ programas (IQ) .....	190
13.27. <u>TRANSSEL</u> - Definir IQ transaccion selecciones (IQ).....	191
14. SYSTEM funciones.....	192
14.1. <u>DEBUG</u> - Encender debug ventana (IQ).....	193
14.2. <u>EXEC</u> - Ejecutar texto como calculo de la linea .....	194
14.3. <u>COGERFLD</u> - Set SY estructura puntos (IQ) .....	195
14.4. <u>INSTALL</u> - Activacion de las funciones externas .....	196
14.5. <u>SYSPAR</u> - Coger sistemaparametro.....	197
14.6. <u>SYSPARSET</u> - Set valor del sistema parametro .....	198
14.7. <u>USERINFO</u> - Coger informacion sobre el usuario.....	199
14.8. <u>WIF</u> - Testmarcar (IQ) .....	200
14.9. <u>WIF</u> - Testmarcar (RAP) .....	201
14.10. <u>WIFS</u> - Testmarcado de los campos (IQ) .....	202
Index.....	203

# 1. Introduccion

La sintaxis de los calculos escritos en RAPGEN esta basado en un BASIC-like lenguaje. Este lenguaje permite probar un valor en el campo, aritmetico declaraciones, texto procesando y mucho mas.

	<b>Reservar palabras</b>	<b>Sinonimos</b>	<b>Descripcion</b>
<b>Condicional declaraciones</b>	IF LET		IF expresion ... IF expresion LET expresion
	ELSE		... IF expresion ... ELSE expresion
<b>Block declaraciones</b>	BEGIN END		empezar bloque fin bloque
<b>Controlar lazo corriente</b>	BREAK CONTINUE		salir de lazo continuar lazo
<b>Operadores logicos</b>	NOT AND OR		no igual a y o
<b>Operadores aritmeticos</b>	+ - * / %		añadir quitar multiplicar dividir porcentage
<b>Relacion operadores</b>	= > < >= <= <>		iguales mas grande que mas pequeño quen mas grande que o igual a menos que o igual a no igual a
<b>Comentario</b>	REM /*		Comentario de la fila lleno comentario despues del estado
<b>Saltar y subrutinas</b>	GOTO GOSUB  RETURN ON..GOSUB..		ir a etiqueta: ejecutar etiquetas de subrutina: volver de subrutina condicional rama

Este lenguaje de las sintaxis suministra con cantidades de caminas a escribir estados. Nosotros ahora cogemos algunos ejemplos de estos:

## **1.1. Ejemplos**

### **1.1.1. IF..ELSE - Condicional estados**

Si el suministrador del balance (LE#6) esta encima 1000 sustrata 100 tambien añadir 47.

### **1.1.2. BEGIN..END - Estados del bloque**

Si el suministrados del balance (LE#6) id encima 1000 cuando empieza el bloque donde 100 es subtraido del balance y linea 7 esta marcado.

Las principales, que todas las lineas entre EMPEZAR y ACABAR estan realizadas unicamente si el condicional es verdad.

### **1.1.3. START/END...NEXT...REPEAT - Lazos**

Los siguientes lazos leidos de todas suministraciones en el rango 111-999. Si el balance es menos que 1000 el suministrados no es procesado.

El siguiente lazo leido de todas las suministraciones en el rango 111-999. Cuando el balance es mas grande que 10000 es encontrado el lazo es acabado.

### **1.1.4. NO, Y, O - Operaciones logicas**

Si la pasada fecha de compra iguales 0 la fecha de compra es set a la fecha de hoy. Estos estamentos iguales

Si el coste de precio no es igual a 0 y la fecha psada de compra no son iguales a 0, cuando marcas la linea 5 en el registro.

Si el precio de coste no son iguales a 0 O la pasada fecha de compra no son iguales a 0, cuando marcas la linea 5 en el registro.

### **1.1.5. REM, /\* - Comentarios**

### **1.1.6. GOTO Saltar etiqueta**

Usando el GOTO estado tu puedes saltar en los calculos normalmente dependiendo en el valor de un campo. Una etiqueta definida como 'NOMBRE:' decide donde salta. En el ejemplo permite la linea 7 es marcada a tres tiempos.

### **1.1.6.1. ON...GOTO/GOSUB - Condicional saltar / subrutinallamada**

Condicional saltar a una etiqueta o llamar a una subrutina dependiendo en el valor en un campo puede ser usada ON. EN puede ser usado ambos con GOTO y GOSUB.

### **1.1.7. GOSUB Llamando subrutinas**

Si los mismos calculos estan hechos a algunos tiempos tu puedes escribir estas lineas como una subrutina expesando con una 'etiqueta:' y llamada con GOSUB

### **1.1.7.1. RETURN** Volviendo de una subrutina

Una subrutina es acabada con VOVLER donde despues los calculos deben ser continuar de donde la llamada cogida en el lugar. Tambien remitir a el VOLVER funcion descritas mas tarde donde un valor puede ser vuelto de los calculos.

## **1.2. Campos**

### **1.2.1. #xx o kk#xx - Campos de una fila**

Tu puedes remitir a un campo de una fila como:

Anotar que kk, KK, Kk y kK remiten diferentes registros de una fila, tu deberias usar normalmente la caja larga kk.

### **1.2.1.1. #xx(de,a) - Parte de los campos**

Parte de los campos estan escritos como kk#xx(de,a) y tu puedes usar esta sintaxis para ambos campos numericos y alfanumericos.

Para un texto de los campos alfanumerico y unico para estos tu puedes tambien asignar un valor a una parte de un campo:

### **1.2.1.2. #xx(no) - Etiquetas de los campos**

Las etiquetas de los campos son referidas como kk#xx(no) donde no esta en el rango 0 hasta max.

Un campo puede ser definido como una tabla del campo en la fecha del diccionario si el formato contiene eg. 20(003) especificamente 3 elementos extra en la tabla o esto puede justamente ser un set de campos contagiosos con el mismo formato los cuales deberian wish to usar como una etiqueta en los calculos. Anotar que los campos libre pueden tambien ser definidos como etiquetas de los campos con el formato especifico.

Un ejemplo de estos es el demo-suministrador de la fila donde el nombre del bloque #2, #3 y #4 tambien puede ser usado como una tabla como #2(0), #2(1) y #2(2)

Anotar que los valores criticos pueden ocurrir si tu exceeds el maximo de una tabla eg. por estar usando #2(4)

### **1.2.1.3. Conversion entre campos numericos y textos**

Tu puedes jsutamente set un campo numerico = un texto del campo como #30 = #2 convertir a numerico y calcular con el valor numerico. Las funciones NUMBER y NUMS pueden ser usadas para mas conversiones avanzadas, ver esto.

En la caja de un campo el texto = un campo numerico como #2 = #30 el resultado debe ser una cuerda del tecto de una longitud variable dependiendo el numero como "123". Normalmente #2 = #30 USING "#####" es usado para especificar la distribucion de los resultados del texto del campo, ver el USING funcion.

## **1.2.2. SY#xx - Sistema de los campos**

Los sistemas de los campos son campos especiales definidos en el pseudofila SY los cuales deben ser siempre presentes. un poco de los sistemas de los campos estan descritos en los siguientes, para una lista completa ver vuestra fila SY de definicion actual.

Un sistema del campo esta remitido a cualquier numero SY#1 o por el nombre corto #DD como el estado en la primera parte del nombre del campo. Algunos de los sistemas del campo estan asociados a una fila y pueden ser cogidos como kk#nombre corto como kk#RECNO

**1.2.2.1. #DD, #PD - Fecha de hoy como la fecha**

Entrar al comienzo de un registro, (99.99.99).

### **1.2.2.2. #PP** - Numero de pagina

Es asignado automaticamente durante el cambio de la pagina, (9999).

### **1.2.2.3. #SN - Nombre del sistema**

Puede ser usado si RAPGEN es instalado con multiples sistemas e.g. diferentes compañías/filas sets. Tambien anotar los campos #SU el nombre contenido del subsystema y #CN con el nombre de la compañía.

### **1.2.2.4. #OK** - Resultados despues de estar leyendo de una fila

Despues de estar leyendo de una fila tu puedes usar #OK, Este campo deberia de ser 0 si un registro estaba leído, alguna cosa tambien indican error.

### **1.2.2.5. #UN** Nombre del usuario

Tu puedes usar #UN coger el nombre del usuario para este PC entrado por la LICENCIA modulo.

**1.2.2.6. #LIN** numero de la linea y **#LOF** lineas en forma

#LIN contiene la actual marca de la linea, #LOF el actual numero de las lineas en forma.

### **1.2.2.7. #LEVEL - Actual nivel total**

Con #LEVEL tu puedes controlar los calculos / marcar dependiendo de el nivel subtotal, ver el RAPGEN usuario manual.

### **1.2.2.8. kk#RECNO - El pasado numero de registro usado de la fila kk**

Si el sistema de la fechabase usado esta conectado con los numeros del registro el pasado uso para la fila kk puede ser encontrado en kk#RECNO. Tambien anotar que los campos kk#NUMBER contienen numeros de registros relativos y kk#NUMBER

### **1.2.3. WW#xx - Campos libre (Campos de trabajo)**

Un programa debe ser creado por estar asignado 40 campos de trabajo los cuales deben de estar definidos cuando el primer tiempo usado y los cuales pueden mas tarde ser cambiados por hacer un dobleclick en el campo.

Los numeros del campo deben de ser una demostracion como una continuacion de los campos en la fila principal pero los campos estan actualmente reservado como WW#1,WW#2,... donde por un cambio mas tarde de el numero de los campos en la fila principal las causad automaticas de renumerar de los campos libres en todos los programas.

El numero de campos libres puede ser ajustado en IQ/DATAMASTER con el parametro del programa de la funcion, en RAPGEN por en los calculos justamente usando un numero mas alto que la demostracion en la lista de la caja las cuales deben de causar el numero de campos libres son extendidas automaticamente.

### **1.2.3.1. #Dntext** - Entrar fecha

En RAPGEN el nombre de un campo libre puede estar empezando con #Dn defines la entrada del campo 1 a 7 son entrados por empezar del registro.

### **1.2.3.2. #Ptexto - Cuadros de los campos**

El nombre de un campo libre empieza con #P y define como texto del campo es una referencia a un cuadro.

## **2. Funciones aritmeticas**

Esta seccion describe las funciones para los calculos numericos iguales como redondear y poder.

## 2.1. **ABS** - El valor absoluto de un numero

numero  $ABS(\text{numero } par1)$

**Parametros:**  $par1$  : numero es convertidos a un valor absoluto

**Descripcion:** La funcion vuelve al valor absoluto de los parametros  $par1$ . Eg. el valor positivo sin signo.

**Volver valor:** El valor positivo.

**Ver tambien:** SGN

**Ejemplo:** #1 =  $ABS(-123.45)$  /\* Campo #1 contiene el valor 123.45

## 2.2. **FNH** - Redondear numero - no decimales

numero FNH(numero *par1*)

**Parametros:** *par1* : define un numero (con decimales)

**Descripcion:** La funcion es usada para redondear un numero con decimales a un numero sin decimales.

**Volver valor:** El numero sin decimales.

**Ver tambien:** FNR, RUN

**Ejemplo:** #1 = FNH(1234.56) /\* Campo #1 contiene el valor 1235

## 2.3. **FNR** - Redondear numero a 2 decimales

numero FNR(numero *par1*)

**Parametros:** *par1* : defines un numero (con decimales)

**Descripcion:** La funcion es usada para redondear un numero con mas que 2 decimales a un numero con unicamente 2 decimales. RAPGEN siempre redondea un resultado a el numero de decimales de los digitos cogidos en el formato del campo. Tu puedes anular esto por estar llamando estas funciones como FNH/FNR.

El estar redondeado puede ser controlado por usar del RUND funcion. Estos definen:

**Vovler valor:** El numero redondeado.

**Ver tambien:** FNH, RUN, RUND

**Ejemplo:** #1 = FNR(123.456) /\* Campo #1 contiene el valor 123.46

## 2.4. **FRA** - Calcular la fraccion de un numero

numero FRA(numero *par1*)

**Parametros:** *par1* : el numero (con decimales)

**Descripcion:** La funcion separa la fraccion del valor de un numero y volver a el.

**Volver valor:** La fraccion como 0.<valor fraccional>.

**Ver tambien:** FNH, FNR, RUN

**Ejemplo:** #1=FRA(123.456) /\* coger 0.456 , #1=FRA(-12.345) /\* coger -0.345

## 2.5. **INT** - El valor entero de un número

numero INT(numero *par1*)

**Parametros:** *par1* : defines un numero

**Descripcion:** La funcion vuelve al valor entero, esto es el valor mas largo y cercano sin decimales.

**Volver valor:** El valor entero.

**ver tambien:** [FRA](#)

**Ejemplo:** #1=INT(1234.56) /\* coger 1234, #1=INT(-12.34) /\* coger -13

## 2.6. **NO** - Negacion logica

numero NO(numero *par1*)

**Parametros:** *par1* : defines un numero

**Descripcion:** La funcion vuelve 1 si la parte1 es igual a zero, 0 si la parte1 no es igual a zero.

**Volver valor:** 0 o 1.

**Ver tambien:** SGN

**Ejemplo:** NO(1) es 0

## 2.7. **POW** - Razon no poder

numero POW(numero *par1*, numero *par2*)

*par2* : defines el exponente

**Descripcion:** La funcion aumenta un numero *par1* a el *par2* poder.

**Volver valor:** El n'th poder.

**Ver tambien:** SQR

**Ejemplo:** #1=POW(8,3) /\* coges 512 (8\*8\*8) , #1=POW(4,0.5) /\* coges 2

## 2.8. **RUN** - Redondeando a x decimales

numero RUN(numero *par1*, numero *par2*)

*par2* : No de decimales a redondear a

**Descripcion:** La funcion de RUN redondear la figura cogida al numero de decimales.

**Volver valor:** La figura redondeada.

**Ver tambien:** [FNH](#), [FNR](#), [INT](#)

**Ejemplo:** #1=RUN(-123.4567,3) /\* Campo 1 ir al valor -123.457

## 2.9. **RUND** - Definición del FNR redondeando la función

numero RUND(numero *par1*, numero *par2*)

*par2* : El numero de decimales redondear A, eg. 2

**Descripción:** El RUND función define como el FNR función están hechos los redondeos. Si *par1* es positivo FNR debería de redondear ARRIBA, si *par1* es negativo FNR debería redondear ABAJO.

**Volver valor:** None.

**Ver también:** [FNR](#)

**Ejemplo:**

## 2.10. **SGN** - Parar si el numero es negativo, zero o positivo

numero SGN(numero *par1*)

**Parametros:** *par1* : defines un numero

**Descripcion:** La funcion examina si el numero es negativo, zero o positivo.

**Volver valor:**

**Ver tambien** INT, NO

**Ejemplo:** #1=SGN(-123.45) /\* Campo #1 cuando contiene el valor -1.

## 2.11. **SQR** - Calcular el cuadrado de la raiz de un numero

numero SQR(numero *par1*)

**Parametros:** *par1* : el numero coge el cuadrado de la raiz de

**Descripcion:** La funcion calcula el cuadrado de la raiz del numero en *par1*.

**Volver valor:** El cuadrado de la raiz.

**Ver tambien:** POW

**Ejemplo:** #1=SQR(4) /\* Coger 2

### **3. Funciones de la cuerda**

Esta seccion describe las funciones para conversiones del texto de los campos y para convirtiendo campos numericos dentro de las cuerdas.

### 3.1. **CONV** - Cambiar caracteres en un texto

texto CONV(texto *par1*, texto *par2*, texto *par3*)

*par3* : los nuevos caracteres son insertados

**Descripcion:** La funcion de los textos de cada caracter en el texto *par1*. Si el caracter iguales a uno de estos en *par2*, esto deberia de deber ser cambiado con el nuevo caracter en *par3*. Si el parametro 1 contiene "abc" y el parametro 2 el texto "ABC", la funcion deberia de reemplazar uno con A, b con B y c con C.

**Volver valor:** El texto donde la contestacion de los caracteres son convertidos.

**Ver tambien:** LOWER, SMAA, UPPER

**Ejemplo:** #1 = CONV("hans", "hn", "lr") /\* Coger "lars"

### 3.2. **EDIT** - Editando de una figura integra

texto EDIT(numero *par1*, texto *par2*)

*par2* : USING la mascara para editar

**Descripcion:** La funcion de editar convierte una figura integra a un texto del campo. El USING mascara determina la distribucion del texto.

**Volver valor:** El texto del campo editado.

**Ver tambien:** NUMBER, USING

**Ejemplo:**

### **3.3. FIND** - Fin del texto en el texto del campo

numero FIND(texto *par1*, texto *par2*, numero *par3*, numero *par4*, numero *par5*)

**Descripcion:** La funcion de buscar para el texto *par1* en el texto *par2*. Ambos parametros tienen que ser cogidos en "" (cuotas).

**Volver valor:** Volver -1 si el texto no es encontrado, otra vision un numero positivo igual a la posicion donde el texto fue encontrado (origen 1).

**Ver tambien:**

**Ejemplo:**

### 3.4. **LEN** - Longitud de un texto

numero LEN(texto *par1*)

**Parametros:** *par1* : defines un texto

**Descripcion:** La longitud de la funcion calculada de un texto.

**Volver valor:** La longitud del texto.

**Ver tambien:** SPOFF

**Ejemplo:**

campo #2 cuando contiene el valor 12, porque hay 12 caracteres en #1.

### **3.5. LOWER** - Convertir el texto a mas bajo que la caja de caracteres

texto LOWER(texto *par1*)

**Parametros:** *par1* : defines un texto que va a ser convertido

**Descripcion:** la funcion de convertir un texto a letras pequeñas, eg. todas las letras A-Z son convertidas a a-z.

**Volver valor:** El texto convertido.

**Ver tambien** CONV, SMAA, UPPER

**Ejemplo:**

### **3.6. NAME** - Extraccion de Christian y nombre pasado

texto NAME(texto *par1*, tal *par2*)

**Descripcion:** La funcion extrae mejos posibles Christiannombre y Pasadonombre de la fuente del nombre cogida y volver al nombre como especificar por *par2*. Este valor puede ser usado para clasificar.

El SSV textofila WORDS.SPA es usado para esto. cada linea contiene una especificapalabra como Mr., Miss, Mrs y otros eventules remplazos (Mister;Mr.)

**Volvervalor:** El nombre como especificar por *par2*.

**Ver tambien:** SMAA, SOGE

**Ejemplo:**

### **3.7. NUMBER** - Conversion de 'sucio' numeros

numero NUMERO(texto *par1*)

**Parametros:** *par1* : Un texto contiene un numero

**Descripcion:** El NUMBER de la funcion extrae un valor de un campo del texto sin estar mirando agunos de los caractes no comienzan con digitos.

**Volvervalor:** Extraer la figura integra, no decimales estan vueltos.

**Ver tambien:** EDIT, NUMS, USING

**Ejemplo:**

### 3.8. **NUMS** - Conversion del numero de texto del campo

numero NUMS(texto *par1*)

**Parametros:** *par1* : Un texto contiene un numero

**Descripcion:** En una linea esta conteniendo #1=#2 donde #1 es numerico y #2 es un textocampo, algun numero en el campo 2 debe de ser convertido automaticamente. El mismo resultado debe de ser usando la extension #1=NUMS(#2) pero NUMS es opcional.

Comosiempre si tu en una linea igual quieres calcular directamente el los valores del campo NUMS deben de ser usados exactamente especificar la conversion como en:  
#1=NUMS(#2)+NUMS(#3)

**Volvervalor:** El valor numerico de el texto del campo. Punto decimal ldebe de ser estado como . (punto)

**Ver tambien:** NUMBER

**EJEMPLO:** #1 = NUMS("aa111") + NUMS("222,22 texto") + NUMS("333.33")

Campo 1 va a la suma de los numeros contenidos en el texto del campo = 555.33

### 3.9. **PACK** - Empaquetacion de un numero

texto PACK(texto *par1*, numero *par2*)

*par2* : 0, no usar, reservar para el futuro tipo de paquete

**Descripcion:** 8870 - basic llamar 60,A\$,B\$ es el mismo como B\$=PACK(A\$)

**Volvervalor:** El valor del paquete del campo.

**Ver tambien:** UNPACK

**Ejemplo:** #1=PACK(#2) /\* #1 ir al valor del paquete de #2

### 3.10. **SMAA** - Convertir texto a mas alto/mas bajo de la caja - nombres

texto SMAA(texto *par1*)

**Parametros:** *par1* : el texto va a ser convertido

**Descripcion:** La funcion de convertir el texto en *par1* mas alto y mas bajo que la caja de las letras. Eg. la primera letra en cada palabra debe de ser set a la caja mas alta mientras el resto estara set a la caja de las letras mas baja. El SSV fila WORDS.SPA deberia de ser parada para ocurrencias de la primera y de la ultima palabra en el texto. Si encuentras el significado de estos deberia de ser cogido de aqui.

Anotar que el SMAA funcion puede ser usado en DATAMASTER tambien para enlinea conversion del nombre de la entrada de los campos.

**Vovlervvalor:** El texto convertido.

**Ver tambien:** CONV, LOWER, NAME, UPPER

**Ejemplo:**

### 3.11. **SOGE** - Creacion de la busqueda de la llave de una direccion del campo

texto SOGE(texto *par1*, numero *par2*)

*par2* : La longitud de una parte del nombre resultado.

**Descripcion:** El nombre de la calle y el numero de la calle es apartado de la direccion cogida del campo. Estos estan cuando combinas dentro de una busqueda de la llave donde el nombre de la llave es fijado de la longitud *par2* preguntar por el numero de la calle. Este campo puede ser usado para clasificar o buscar.

**Volvervalor:** La longitud del nombre de la calle *par2* preguntar por los 4 digitos del numero de la calle.

**Ver tambien:** LOWER, NAME, SMAA, UPPER

**Ejemplo:**

### **3.12. SPOFF - Remover delante y detras de los espacios en el texto**

texto SPOFF(texto *par1*, Bitflag *par2*)

**Descripcion:** La funcion remueve todos los espacios delanteros y traseros. Mas lejos esto reduce todas las posiciones en blanco a un maximo de un unico espacio en el caracter.

**Volvervalor:** El texto convertido.

**Ver tambien:** LEN

**Ejemplo:**

### 3.13. **UNPACK** - Desempaquetacion de un numero

texto UNPACK(texto *par1*, numero *par2*)

*par2* : 0, no usar, reservar para el futuro tipo de paquete

**Descripcion:** 8870 - basic llamar 61,A\$,B\$ es el mismo como B\$=DESEMPAQUETAR(A\$)

**Volver valor:** El desempaqueado del valor del campo.

**Ver tambien:** PACK

**Ejemplo:** #1=UNPACK(#2) /\* #1 ir al valor desempaqueado de #2

### **3.14. UPPER** - Convertir el texto en el mas alto de la caja

texto UPPER(texto *par1*)

**Parametros:** *par1* : defines un texto que va a ser convertido

**Descripcion:** la funcion de convertir un texto a la caja mas alta, eg. todas las letras a-z son convertidas a A-Z.

**Volver valor:** El texto convertido.

**Ver tambien:** CONV, LOWER, SMAA

**Ejemplo:**

### 3.15. **USING** - Editando un numero

texto USING(numero *par1*, texto *par2*)

*par2* : USING la mascara para estar editando

**Descripcion:** USING la funcion de convertir un numero a un texto del campo. USING la mascara de determinadas distribuciones del texto.

La funcion puede ser llamada con el especial BASIC sintaxis tambien como: textocampo = numero USING "mascara"

**Vol**~~v~~**er**valor: Editar texto del campo.

**Ver tambien:** EDIT

**Ejemplo:**

## **4. Dígito parado y valor**

Esta sección describe las funciones para el dígito parado del cálculo y valor del texto y números.

## 4.1. **CCODE** - Texto del campo parado (DATAMASTER codigo del texto parado)

texto CCODE(texto *par1*, campo *par2*)

*par2* : Numero del campo con parada de los definidos como "7", "#7", "va#7", "va07"

**Descripcion:** La funcion de leer el campo definido de la Fecha Diccionario para el campo cogido *par2* y encuentra los codigos parados definidos para estos. El texto conectado con el valor cogido en *par1* es volver.

**Volvervalor:** El texto parado. Espacios indicados no permitidos, "-" no definicion parada.

**Ver tambien:** VALID, VALCH

**Ejemplo:** #1 = CCODE(9,"va#7") /\* Coges "Especial"

## 4.2. **CODE** - OCR parada

texto CODE(texto *par1*)

**Parametros:** *par1* : es un numero como el numero del cliente

**Descripcion:** Los procesos de la funcion de un numero y volver un texto contenido un OCR valorparado.

#47=CODE (#19) deberia de calcular el OCR digito parado modulos 10 con peso 212121... para el texto del campo #19 y ads esto como el digito pasado.

CODE("123456789012345") volver un texto con un caracter añadido: "1234567890123452".

**Volvervalor:** La suma del texto el OCR digito parado.

**ver tambien:** CHEX

**Ejemplo:** #1 = CODE("33330556") /\* Coges "333305563"

### 4.3. **CHEX** - Modulo 11 parado

texto CHEX(texto *par1*, texto *par1*)

*par2* : Peso para los calculos de los digitos parados, 2 digitos para cada entrada de los caracteres

**Descripcion:** #47=CHEX (#15,"01020304") deberia como CODE el calculo a un digito parado y añadir esto al volver al campo.

El digito parado es calculado usando modulos 11 con los pesos 01, 02, 03, 04 de acuerdo al segundo parametro. Cada set de 2-digitos en este parametro coges el peso para un digito en el parametro 1 del campo.

**Volver valor:** la suma del texto de los digitos parados.

**ver tambien:** CODE

**Ejemplo:** #2=CHEX("330556", "010203040506") /\* coges "3305569"

#### 4.4. **VALCH** - Parar si el texto encontrado en el valor del rango

numero VALCH(texto *par1*, texto *par2*)

*par2* : los valores permitido separados con coma

**Descripcion:** La funcion valida *par1* encontrada entre los valores cogidos en *par2*. Todos los valores cogidos en *par2* tienen que ser separados con , (coma).

**Vovlervvalor:** Volver 0 si *par1* no encontrado en *par2*.

**Ver tambien:** CCODE, VALID

**Ejemplo:** #1=VALCH("Chris", "Anne,Nette,Chris,Ole,Michael") /\* #1 cuando contiene el valor 2.

## 4.5. **VALID** - Parar si el numero encontrado en los numeros validos

numero VALID(numero *par1*, numero *par2*, numero *par3*)

. **Descripcion:** La funcion de los valores si el valor en *par1* es permitido por estar parando los valores permitidos en *par2*. La sintaxis para *par2* es:

"1,2,8-10,12" Estos son los valores 1, 2, 8 a 10 y 12 estan permitidos.

"-1,2,8-10,12" Si un menos es el primer caracter los valores no estan permitidos.

deberia de cambiar el valor del rango del campo #20 por estar entrando 15 tambien #20 ir:  
"1-3,8-12,15"

**Volvervalor:** Volver 0 si *par1* no encontrado en *par2*.

**ver tambien:** CCODE, VALCH

**Ejemplo:** #1 = VALID(9, "1,2,8-10,12")

Campo #1 cuando contiene el valor 3 como el valor es encontrado dentro del tercer rango.

## **5. Manipulacion de las funciones de la fecha**

Calculo de la fecha es un sience para ellos y es descrito en este capitulo.

## **5.1. DATE** - Fecha actual YYYYMMDD

numero DATE()

**Vovler valor:** la fecha actual como YYYYMMDD.

## 5.2. **DATECALC** - Calcular una fecha

Fecha DATECALC(Fecha *par1*, numero *par2*, numero *par3*, numero *par4*, numero *par5*)

*par5* : dia(s) DD

**Descripcion:** La funcion puede ser usada a set una fecha, o añadir a/sacar de una fecha. Si *par2* es set a 0 una fecha puede ser usando los parametros *par3-par5*. Si el parametro 3, 4 y 5 estan set, el parametro 1 deberia ser ignorado. To set el unico mes, la funcion usa la fecha en *par1* y cambiar el mesa a el uno en *par4*.

**Volver valor:** Los calculos de la fecha como YYYYMMDD.

**Ver tambien:** DIA, FNA, FNB, FND, FNU, FNV, FNY, MES, WDIA, WORKD

**Ejemplo:**

### 5.3. **DIA** - Descripcion de una fecha - textoforma

texto DIA(Dia *par1*)

**Parametros:** *par1* : una fecha como YYYYMMDD

**Descripcion:** La funcion crea un texto con la fecha como: <?> <diasemana> el. <dia> <mes> <año>

Si el fia es un 'libre-dia' el primer caracter deberia ser un \*, si unico un 'pasado libre-dia' un /, otravision en blanco. El mismo calendario como describir para TRABAJAR esta usado.

**Volver valor:** Volver un texto.

**ver tambien:** DATECALC, FNA, FNB, FND, FNU, FNV, MES, WDIA, WORKD

**Ejemplo:** #1 = DIA(19931016) /\* crear el texto para 16. octubre 1993

Campo #1 contiene el valor "\*El sabado 16 de octubre 1993"

## 5.4. **FNA** - Convertir fecha a un numero de dias de año 0

numero FNA(Fecha *par1*, numero *par2*)

**Descripcion:** La funcion de los calculos estan cogidos a la fecha a el numero de dias desde el año 0. Este valor puede ser usado a añadir o sacar dias o calcular la diferencia entre las fechas.

**Volvervalor:** El numero de dias desde el año 0.

**Ver tambien:** FNB, FND, FNU, FNV, DATECALC, DIA, MES, WDIA, WORKD

**Ejemplo:**

Campo #2 contiene el valor 728277 y campo #3 el valor 13

## 5.5. **FNB** - Convertir numero de dias de año 0 a fecha

Fecha FNB(numero *par1*, numero *par2*)

**Descripcion:** Calcular la funcion una fecha YYYYMMDD en bases de un valor. Eg. un numero vuelto de la funcion FNA() puede ser analizado como parametro a estas funciones y siempre volver a una fecha valida.

**Volvervalor:** Volver el valor como una fecha YYYYMMDD.

**ver tambien:** DATECALC, DIA, FNA, FND, FNU, FNV, MES, WDIA, WORKD

**Ejemplo:**

Campo #2 contienen el valor 19931224, eg. 24. diciembre 1993

## 5.6. **FND** - Fecha conversion

Fecha FND(Fecha *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD

**Descripcion:** Esta funcion puede ser usada a convertir fechas de un formato a otro, y es normalmente usado con clasificacion y selecciones. Ex.

**970101 es mas grande que 961231 pero 311296 es mas grande que 010197**

Tu puedes ver lo necesitado para estar usando el FND funcion si tu probar similar comparencias con una fecha del campo definido DDMMYY.

**Volver valor:** Vuelves el valor como una fecha YYMMDD o DDMMYY.

**ver tambien:** DATECALC, DIA, FNA, FNO, FNU, FNV, FNY, MES, WDIA, WORKD

**Ejemplo:**

## 5.7. **FNE** - Convertir la fecha a un numero del mes

numero FNE(*fecha par1*)

**Parametros:** *par1* : Una fecha como YYYYMMDD o YYMMDD

**Descripcion:** Esta funcion puede ser usado a calcular fechas diferentes en meses.

**Volver valor:** La funcion calculada el numero de mes como Año\*12 + Mes (YY\*12+MM)

**Ver tambien:** [DATECALC](#), [DIA](#), [FNA](#), [FNB](#), [FND](#), [FNV](#), [MES](#), [WDIA](#), [WORKD](#)

**Ejemplo:** #1 = FNE(19950331) /\* *coges 1143 = 95\*12+03*

## 5.8. **FNF** - Convertir fecha a numero del dia, 360 dias/año

numero FNF(Fecha *par1*)

**Parametros:** *par1* : Una fecha como YYYYMMDD o YYMMDD

**Descripcion:** Esta funcion calcula el numero del dia del año 0 usando 360 dias/año. Mismo como FNA(fecha,360)

**Volver valor:** Numero de dias de año 0.

**Ver tambien:** [FNA](#)

**Ejemplo:**

## 5.9. **FNO** - Convertir fecha a DDMMYY

Fecha FNO(Fecha *par1*)

**Parametros:** *par1* : Coger fecha como DDMMYY, YYMMDD o YYYYMMDD

**Descripcion:** Nomateria como la entrada de la fecha es entrada a la fecha deberia ser vuelto como DDMMYY. Esto puede ser usado cuando en una subsecuencia marcada fuera.

**Volver valor:** DDMMYY

**Ver tambien:** FND, FNY

**Ejemplo:**

## 5.10. **FNU** - Convertir fecha a dia de la semana

numero FNU(Fecha *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD

**Descripcion:** La funcion es usada a calcular el dia de la semana de una fecha.

**ver tambien:** DATECALC, DIA, FNA, FNB, FND, FNV, MES, WDIA, WORKD

**Ejemplo:** #1 = FNU(19931215) /\* que dia es 15. diciembre 1993 ?

campo #1 contiene el valor 4 (=Miercoles)

## 5.11. **FNV** - Convertir fecha a no de la semana a fecha

numero FNV(numero *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD, o un numero de la semana como YYYYWW

**Descripcion:** La funcion convierte a fecha a el numero de la semana YYYYWW, si *par1* es una fecha. Si *par1* en la otra mano es un numero de la semana YYYYWW la funcion deberia volver a una fecha igual a el domingo pasado primero coger la semana. Mismo como SEMANA(fecha)

**Volvervalor:** Volver un numero YYYYWW, donde YYYY = año y WW = ugenr, o una fecha YYYYMMDD.

**Ver tambien:** DATECALC, DIA, FNA, FNB, FND, FNU, MES, WDIA, SEMANA, WORKD

**Ejemplo:**

Campo #1 cuando contiene el valor 199341, igual al numero de la semana 41. Campo #2 contiene la fecha 19931010.

## 5.12. **FNY** - Convertir fecha a YYYYMMDD

Fecha FNY(Fecha *par1*)

**Parametros:** *par1* : Coger fecha como DDMMYY, YYMMDD o YYYYMMDD

**Descripcion:** Nomateria como la entrada de la fecha es vuelta a la fecha deberia ser vuelta como YYYYMMDD. Esto puede ser usado cuando en una subsecuencia de los calculos.

**Volver valor:** YYYYMMDD

**ver tambien:** FND, FNO

**Ejemplo:**

### 5.13. **MES** - texto generado con descripcion de un mes

texto MES(Fecha *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD

**Descripcion:** la funcion genera un texto igual a el nombre del mes necesitado.

**Volvervalor:** Volver el nombre del mes.

**Ver tambien:** DATECALC, DIA, FNA, FNB, FND, FNU, FNV, WDIA, WORKD

**Ejemplo:** #1 = MES(19931016) /\* fecha es 16. octubre 1993

campo #1 cuando contiene el valor "octubre".

## **5.14. TIEMPO** - Tiempo actual TTMMSS

numero TIEMPO)

**Volver valor:** El tiempo actual como TTMMSS.

## 5.15. **WDIA** - Describir el dia de la semana de la fecha

texto WDIA(Fecha *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD

**Descripcion:** la funcion genera un texto como: <?> dias de la semana

Si el dia es un libre-dia el primer caracter deberia ser un \* u esto es un / si el unico pasado libre-dia. Otravision en blanco. El mismo calendaria como descripcion para WORKD es usada.

**Volver valor:** Un texto con el dia.

**ver tambien:** DATECALC, FNA, FNB, FND, FNU, FNV, MES, WDIA, WORKD

**Ejemplo:** #1 = WDIA(19931016) /\* Campo #1 contiene el valor *"\*Sabado"*

## 5.16. **SEMANA** - Convertir fecha a semanano o semanano a fecha

numero SEMANA(numero *par1*)

**Parametros:** *par1* : defines una fecha como YYYYMMDD, o un numero de la semana como YYYYWW

**Descripcion:** La funcion convierte una fecha a un numero de la semana YYYYWW, si *par1* es una fecha. Si *par1* en la otra mano es un numero de la semana YYYYWW la funcion deberia volver a la fecha igual a el pasado domingo primero coger la semana. Mismo como FNV(fecha)

**Volvervalor:** Vuelve un numero YYYYWW, donde YYYY = año y WW = ugenr, o una fecha YYYYMMDD.

**ver tambien:** [FNV](#)

**Ejemplo:**

Campo #1 cuando contiene el valor 199341, igual a numero de la semana 41. Campo #2 contiene la fecha 19931010.

## 5.17. **WORKD** - Calcular el numero de dias trabajados entre las fechas

numero WORKD(fecha *par1*, Fecha *par2*)

*par2* : defines una fecha como YYYYMMDD

**Descripcion:** La funcion calcula el numero de semanas de dias de trabajo entre dos fechas.

#47 = WORKD (#15,#PD) calcular el numero de dias de trabajo actuales de la fecha en el campo 15 a la fecha entrada en 'como fecha de'.

la funcion empieza por estar calculando el numero de dias entre las dos fechas. Todos los sabados y domingos deberian de ser sacados. Como el paso final las funciones buscada un 'dia de trabajo en la fila de la tabla', donde las vacaciones estan listadas, y cuando sacas a llenar o pasado dia por dia encontrado.

Esta fila de la tabla puede si necesariamente ser ajustada individualmente. la funcion usada la fila RAPDAY.SPA. Esta fila es un SSV tekstfila donde cada linea contiene unas vacaciones como YYYYMMDD. poara vaciones pasadas el siguiente porcentaje de libredom como eg. 19960630;50

**Volvervalor:** volver el numero de dias de trabajo entre fechas.

**ver tambien:** DATECALC, FNA, FNB, FND, FNU, FNV, MES, WDIA, WORKD

**Ejemplo:** #1 = WORKD(19930420, 19930430) /\* Campo #1 cuando contiene el valor 19.

## **6. Mano de multiples campos**

Este capitulo describe funciones para manejar un manajo de campos, especialmente el LET funcion.

## 6.1. **LET** - Calculando varios campos a tiempo

numero LET(campos *par1*)

**Parametros:** *par1* : defines uno o mas campos

**Descripcion:** la funcion es usada a calcular uno o mas campos usando un estado. Los campos pueden calcular con la expresion campos **XX** constante/campo, donde **XX** puede ser

<b>Operador</b>	<b>Principal</b>
=	set campos iguales a
+=	añadir valor a los campos
-=	restar valor de los campos
*=	multiplicar campos con el valor
/=	dividir campos con el valor
%=	set campos a el mod. valor de la division
&=	necesitas una operacion logica y en los campos
=	necesitas logica o operacion en campos

**Volvervalor:** volver 0 si el calculo fuese afortunado.

**Ver tambien:** CLEAR, ZERO

**Ejemplo:**

### **Letexpresion**

LET("#1-10=12")

LET("#20,25=3,7")

LET("#20-25=le#1-10")

LET("#20-25=le#1-2")

LET("le#1,3,va#7=#1,ku#3")

LET("#20-25+=1")

### **Funcion**

Campo 1 a 10 es set igual a 12

#20=3 y #25=7

campo 20-25 es set a la fila del campo 1-6

#20=#22=#24=le#1, #21=#23=#25=le#2

Algunas filas pueden ser mixtas

Añadir 1 a todos los campos 20-25

### 6.1.1. **LET** - Asignar valores a los campos entre IQ programas (IQ)

numero LET(campos *par1*)

**Parametros:** *par1* : defines uno o mas campos

**Descripcion:** El LET asignado de multiples campos tienen que ser extendidos a trabajar con multiprogramas y entre lineas en lista/transaccion programas.

**Volvervalor:** Volver 0 si el calculo era afortunado.

**Ver tambien:**

**Ejemplo:**

**Letexpresion**

LET (20.#1-3=#1-3)

LET (#1-3=20.#4-6)

LET (#10=#3.4)

**Funcion**

Sets campo 1-3 para programa 20 = este programa #1-3

Sets campo 1-3 en este programa a#4-6 de programa 20

Sets campo 10 igual a campo 3 de linea 4

## 6.1.2. LET - Creando nuevas filas (RAP)

numero LET(campos *par1*)

**Parametros:** *par1* : defines uno o mas campos

**Descripcion:** El LET funcion puede ser usado para construir nuevas filas.

**Volvervalor:** Volver 0 si el caculos era afortunado.

**Ver tambien:** INSERT, UPDATE, *Rapgen Manual*

**Ejemplo:**

### **Letexpresion**

LET (aa=#1-3,87,le#2)

LET (aa=#1-3,6K,15D)

LET (aa=#1-3,6,15:2,NP)

LET (aa=#1-3),12000

LET (aa=#1-3),-1

LET (aa=#1-3),1000,xnet

LET (aa=#1-3) -acc

LET (07/aa=#1-3),25

### **Funcion**

Define fila aa, llave=aa#1, tipo=1.fechabase conductor

Llaves aa#4 y aa#5 (duplicar)

Llaves aa#2 y rel.recno (duplicar)

12000 registros (falta es 1000 si necesitas)

Fila deberia de ser construida a cada tiempo

Fila es un XNET fila

Fila es un acceso a la fila, construir siempre

Lu puede ser cogido para filas basicas

## 6.2. **CLEAR** - tocar todos los campos en una fila (RAP)

numero CLEAR(fila *par1*)

**Parametros:** *par1* : el nombre corto de la fila

**Descripcion:** La funcion sets todos los campos para una fila a zero.

**Volver valor:** volver 0 si ok.

**ver tambien:** ZERO

**Ejemplo:**

El ejemplo inserta un nuevo registro en el articulo de la fila. Debido a la funcion CLEAR() todos los otros campos que el numero de articulo estan set a zero.

### **6.3. CLRFLAG** - Limpiar opciones para los campos en la pantalla (IQ)

CLRFLAG(campos *par1*, numero *par2*, numero *par3*)

**Descripcion:** Cada campo de la pantalla es asociado con parametros (bits) definiendo el uso. El SETFLAG funcion puede ser usado a set estas banderas, CLRFLAG limpiarlos.

**Ver tambien:** SETFLAG, GETFLAG

**Ejemplo:** CLRFLAG("#12,44",7,0)

## 6.4. **COLOR** - Set background caja de color para un numero de campos

COLOR(campos *par1*, Colorrojo *par2*, Colorverde *par3*, Colorazul *par4*)

*par4* : Valor de color azul (0-255)

**Descripcion:** El background color para los campos cogidos estan set a el RGB valor, esto es el campo de la caja es llenar con el color cogido.

**Volvervalor:** None

**Ver tambien:** COLORF

**Ejemplo:**

## 6.5. **COLORF** - Set foreground texto color para un numero de campos

`COLORF(campos par1, ColorRojo par2, ColorVerde par3, ColorAzul par4)`

*par4* : Valor de color azul (0-255)

**Descripcion:** El foreground color para coger los campos que son set a el RGB valor, este es el texto del campo que esta marcado en el color escogido.

**Volvervalor:** None

**Ver tambien:** COLOR

**Ejemplo:** `COLORF("#3-4",0,0,255)`     */\* Campo 3 y 4 estan marcados en azul*

## **6.6. DIALOG - Funcion adicional para la entrada**

Numero DIALOG(campos *par1*)

**Parametros:** *Par1*: Campos demostrados en el dialogo

**Descripcion:** El DIALOG funcion permite al usuario a pop arriba el las cajas del dialogo con una selecta set de los campos y algunos puntos de un registro ejecutado o en un IQ programa por ejemplo por hacer un click en un campo.

DIALOG("#1,7-8,le#3") defines un dialogo con los campos cogidos. La documentacion de los campos estan usados como reserva en la linea de la ayuda cuando el cursor del raton es movido encima del texto principal.

Siempre con un campo puede estar una de las opciones siguientes añadidas:

**Volver valor:** OK=0, CANCEL=1

**Ver tambien:** PARAMS

**Ejemplo:**

## **6.7. GETFLAG**- Poner opciones para los campos en la pantalla (IQ)

numero GETFLAG(campos *par1*, numero *par2*, numero *par3*)

**Descripcion:** Cada campo de pantalla esta asociado con parametros (bits) definiendo el uso. El SETFLAG funcion puede ser usado a set estas banderas, CLRFLAG limpiarlos. El GETFLAG funcion puede ser usada para leer estas banderas.

**Volver valor:** None

**Ver tambien:** SETFLAG, CLRFLAG

**Ejemplo:** GETFLAG("#12,44",7,0)

## **6.8. SETFLAG**- Set opciones para campos en pantalla (IQ)

SETFLAG(campos *par1*, Bitbandera *par2*, numero *par3*)

**Descripcion:** Cada campo de la pantalla esta asociado con parametros (bits) definiendo el uso. El SETFLAG funcion puede ser usada a set estas banderas, CLRFLAG limpiarlas. Para el tipo de parametro 0 unicamente deberia ser usado normalmente.

**volvvalor:** None

**Ver tambien:** GETFLAG, CLRFLAG

**Ejemplo:** SETFLAG("#12,44",7,0)

## 6.9. **ZERO** - Zero un numero de los campos

ZERO(campos *par1*)

**Parametros:** *par1* : Especificacion del campo

**Descripcion:** Los campos cogidos son zeroed. ZERO esta trabajanto justamente con la funcion LET.

**Vovler valor:** None

**Ver tambien:** LET, CLEAR

**Ejemplo:** ZERO("3,19")     /\* Zeroes del campo 3 y campo 19

## **7. control del registro**

El capitulo describe las funciones para controlar la corriente del registro calculaos/marcas en RAPGEN. Las funciones CHAIN, MESS y RETURN puede ser tambien usado en IQ y DATAMASTER, las otras funciones estan de no interesantes para los programas de la pantalla.

## 7.1. **CHAIN** - Empezar el proximo registro o otro programa(RAP)

numero CHAIN()

*par3* : Blanco o Indice,Niveltotal,Numero de compañía

**Descripcion:** CHAIN(7) empezar el numero del registro 7 cuando este registro esta acabado.

Al empezar los mismos parametros como para este registro deberia de ser usado.

CHAIN(7,"310395,-,9999","1") sets Tambienfecha a 310395, Empezarllave a nada,Pararllave a 9999 y bajar el nivel total a 1. Los otros parametros al empezar quedan incambiables.

CHAIN(2007) empezar registro numero 7 en subsistema 2.

CHAIN(-1,"c:/windows/write.exe") debe de empezar esta (windows)programa.

Cada tiempo CHAIN es ejecutado un nuevo numero corrido es cogido para empezar de 1 y hacia delante. Un registro es empezado del menu que tiene el numero corrido 0. Tu puedes usar #20=CHAIN() sin parametros para CHAIN coger este numero corrido y hacer un reregistro con un numero corrido a tiempo, eg. marcar un numero de copias.

CHAIN("c:/windows/write.exe") puede ser usado en IQ/DATAMASTER programas empezar otras ventanans del programa.

**Volvervalor:** CHAIN() volver al actual numero corrido.

**Ver tambien:** EXIT , CHAINR

**Ejemplo:**

### **7.1.1. CHAINR** - Cadena del programa o comando externo directamente (RAP)

CHAINR(numero *par1*, texto *par2*, texto *par3*)

*par3* : Blanco o Indice,Niveltotal,Numero decompañia

**Descripcion:** La CHAINR comando debe ser siempre reemplazado PASADA este es el proximo programa debe de ser empezado despues de acabarlo.

Usar CHAINR en lugar de CHAINR interrumpir este programa y llamar -arriba otro programa inmediatamente.

**Volvalor:** None

**Ver tambien:** EXIT , CHAIN

**Ejemplo:** CHAINR(-1,"Bloc") /\* Empezar bloc a la derecha ahora

## **7.1.2. CHAIN** - Cadena IQ programa o comando externo (IQ)

CHAIN(texto *par1*, texto *par2*)

*par2* : Llave opcional para registro to display

**Descripcion:** Activar un numero del programa o un comando de las cuerdas de las ventanas.

**Volvervalor:** None

**Ver tambien:** EXIT, ISACTIVE, WAIT

**Ejemplo:**

## 7.2. **WAIT** - Esperar para el programa acabar (IQ)

WAIT(programno *par1*)

**Parametros:** *par1* : Numero del programa

**Descripcion:** Esperar para coger programa acabado (ver EXIT). Calculos deben de continuar cuando el programa de la ventana esta cerrado.

**Volver valor:** None.

**ver tambien:** CHAIN , EXIT

**Ejemplo:** WAIT(20) /\* no debes de continuar antes que el programa sea leido

### **7.3. COMPILAR** - Compilar un registro (RAP)

COMPILAR(numero *par1*)

**Requisitos:** Es el unico posible de usar esta funcion si un C compila esta instalado y RAPGEN esta comprado con las licencias para compilar.

**Descripcion:** En lugar de seleccionar 'Compilar' del 'Parametro' menu donde siempre el registro esta empezado despues de modificar este puede ser fijado en los calculos.

**Ver tambien:** INSTALL

**Ejemplo:** COMPILAR /\* El registro debe de ser compilado

## **7.4. EXIT - Salida del registro (RAP)**

numero EXIT(numero *par1*)

**Descripcion:** La funcion termina el registro o el paso actual (clase/marcar).

**Volvervalor:** None

**Ver tambien:** CHAIN , CHAINR , MESS

**Ejemplo:**

### 7.4.1. **EXIT** - Cerrar IQ programa o ventana (IQ)

EXIT(numero *par1*)

**Parametros:** *par1* : Numero de programa cerrado

**Descripcion:** EXIT(0) cierras el actual IQ programa.

**Volver valor:** None

**Ver tambien:** CHAIN , MESS, WAIT

**Ejemplo:**

## 7.5. **KEYS** - Externa empezar/parar rangos (RAP)

numero KEYS()

*par2* : Fijar nombre eventual para .LLAVE definicion fila

**Descripcion:** Usando las KEYS funcion tu puedes hacer un registro corriendo con un numero de empezar/parar rangos definidos como lineas en un texto externo de la fila. KEYS cuando reemplazas la entrada de EMPEZAR/PARAR llaves por empezar del registro y puede tambien reemplazar el INDICE especificado.

las llaves de la fila pueden ser creadas con algun editor de texto y algunas lineas contenidas:

Cada linea puede contener unas unicas llaves o llave para los rangos marcados. 2: especiales usos del indice 2, 47/ estado de codigo del calculo los cuales pueden leer en los calculos con #20=KEYS() y usar para los calculos individuales.

Uso de LALVES(0) produce una lista que tienen contenidos todos los registros especificos en la fila de las llaves, KEYS(1) produce una lista separada para cada linea en las llaves de la fila y el ENDSUM rutina puede ser usado para coger un total de estos registros.

Tu puedes controlar un registro con unas llaves de la fila tambien sin reemplazar unos calculos de las KEYS. Por empezar de alguna lista, en EMPEZAR DE, tu puedes entrar:

Si camino/extension esta omitida para las llaves de la fila esto deberia ser cogido del directorio del registro normal con la extension .LLAVE, eg. c:/rapfil/rap/aa.key

**Volvelvalor:** KEYS() vuelve los codigos calculados (47 de 47/111-222) para el rango actual.

**Ver tambien:** ENDSUM, INDICE

**Ejemplo:**

## 7.6. **INDICE** - Set indice y empezar/parar valor para el registro (RAP)

numero INDICE(indICE *par1*, textO *par2*, textO *par3*)

*par3* : el valor que el usuario normalmente entra en el campo parado a

**Descripcion:** La funcion es usada para hacer cumplir un indice y empezar/parar rango para un registro. Si *par1* >= 1 el indice es set para la fila principal, eg. en cuales la orden del registro deberia leer los registros. Si *par2* contiene alguna cosa la funcion debe de ser al empezar el rango y los cuales pone tambien para *par3*.

Si al empezar/parar parametros tienes el primer caracter como suma (+) el valor es reemplazado en frente de alguna entrada hecha en el comienzo/parar de los campos por empezar de el registro.

INDICE(-2) cerraduras del registro para usar el indice 2 pero en orden descendiente. La fechabase conductora debe de soportar lectura descendiente.

**Volver valor:** Volver el indice de la fila principal si debe de ser usado.

**Ver tambien:** KEYS

**Ejemplo:** INDICE(2,"D","D") /\* los registros de la fila principal son KU (fila actual)

El ejemplo hace cumplir al indice 2 para el registro, tambien que los actuales son clasificados de acuerdo al nombre actual y no al codigo actual. Mas lejos leer unicamente los registros, donde el nombre actual empieza con la letra "D".

INDICE(1,"+02","+02") /\* Marcar 024711 arriba en la entrada 4711

## 7.7. LTOT - el nivel total mas bajo (RAP)

numero LTOT(nivel *par1*)

**Parametros:** *par1* : el nivel total mas bajo necesitado para el registro

**Descripcion:** Si *par1*  $\geq 0$  la funcion sets el nivel total mas bajo para el registro. Este nivel igual a uno el usuario normalmente selecciona cuando esta empezando un registro.

**Volvervalor:** Volver el nivel total mas bajo de los registros.

**Ver tambien:** MTOT

**Ejemplo:** LTOT(1) /\* *marcar totales unicamente, apretar todas las especificaciones*

## 7.8. **MTOT** - Nivel total mas alto (RAP)

numero MTOT(nivel *par1*)

**Parametros:** *par1* : el nivel total mas alto para el registro

**Descripcion:** La funcion hace cumplir el nivel total mas alto del registro. Si *par1* iguales 0, el registro no debe de marcar algunos totales.

**Volvervalor:** Volver el nivel total mas alto.

**ver tambien:** LTOT

**Ejemplo:** MTOT(1)     /\* Un non-relevante grande total empieza al apretarlo

## 7.9. **MESS** - Display mensaje en pantalla

numero MESS(texto *par1*)

**Parametros:** *par1* : el mensaje es displayed

**Descripcion:** MESS displays el texto en un mensaje de la caja de Windows. Dependiendo en el pasado caracter en el texto el simbolo siguiente y botones estan usados:

<b>texto</b>	<b>Simbolo</b>	<b>Botones</b>	<b>Fallodelboton</b>
texto	Info	OK	OK
texto?	!	OK, CANCELAR	CANCELAR
texto??	?	SI, NO, CANCELAR	SI
texto!	!	SI, NO	SI
texto!!	PARAR	OK	OK
texto?!	PARAR	OK, PUEDE	OK

**Volvervalor:** 0=OK o SI, 1=NO, -1=CANCELAR

**Ver tambien:** EXIT

**Ejemplo:**

## **7.10. NOPAS** - No contraseña/nombre del usuario en este registro (RAP)

NOPAS()

**Parametros:** None

**Descripcion:** La funcion mueve algunas contraseñas protegidas del registro. Normalmente una actualizacion del registro automaticamente coge la contraseña CARE. Usando NOPAS() o PAS() esta contraseña puede ser movida o cambiada.

**Ver tambien:** PAS, UPDATE

**Ejemplo:**

## 7.11. **PAS** - Set contraseña/nombreusuario (RAP)

numero PAS(text *par1*)

**Parametros:** *par1* : la solicitud de la contraseña/nombreusuario

**Descripcion:** La funcion hace cumplir una contraseña/nombreusuario para un registro. Esta contraseña esta cunado necesita otro registro para empezar.

**Vertambien:** NOPAS

**Ejemplo:** PAS("SWTOOLS") /\* set contraseña a SWTOOLS

## **7.12. PARAMS** - Funcion para añadir registro al empezar los parametros (RAP)

PARAMS(Campos *par1*)

**Parametros:** *Par1*: Campos demostrar al empezar el dialogo en los parametros

**Descripcion:** PARAMS("#1,7-8,le#3") es una variante del dialogo de la funcion donde la entrada esta hecha para empezar del registro que no esta durante la ejecucion del registro.

Uso de los PARAMS en un registro que deberia de añadir un boton <parametros extra> a el empezar encima de la pantalla los cuales cuando activas el dialogo.

**Volvervalor:** None.

**Ver tambien:** DIALOG

**Ejemplo:**

## 7.13. **RETURN** - Volver para los calculos

numero RETURN(numero *par1*)

**Parametros:** *par1* : el valor debe de ser retornado

**Descripcion:** La funcion esta usada a salir de los calculos necesitados para el actual registro de la fila principal. Si no parametros estan cogidos o *par1* iguales 0, el registro debe de ser marcado al definir la marca de las lineas para el registro. Si el valor es non-zero el registro no debe de ser procesado a ningun nuevo o marcado.

**Volvervalor:** None.

**Ver tambien:** GOSUB

**Ejemplo:** IF LE#6 < 1000 RETURN(1) /\* no marcar si el balance < 1000

## **7.14. SORTKEY - Insertando clase de llave adicional (RAP)**

numero LLAVECLASE(ficha *par1*)

**Parametros:** *par1* : 0, -1 o ficha

**Descripcion:** En alguna caja especial una lista deberia de ser clasificada marcando el mismo registro multiplicado por los articulos en la salida. Por ejemplo una lista de articulo donde el articulo esta hecho para encontrar con el numero normal del distribuidor y la alternativa del numero del distribuidor si ninguno.

En la clase de la caja tu deberias de clasificar usando un trabajo del campo los cuales cuando estan calculados y una clase de llave extra esta relacionada donde nunca la funcion de la SORTKEY es llamada.

Algunas filas pueden tambien ser unidas usando esta funcion. La clasetrabajofila contiene un numero normalmente punteado a un registro del registro de la fila principal. Con SORTKEY (le) un registro es insertado punteado a la fila le y con #20= SORTKEY(-1) el numero de campo de la fila actualmente empieza la fila principal es devuelta las cuales pueden ser usadas para controlar mas calculos.

**Volvervalor:** Numero de fila principal, normalmente 1.

**Ver tambien:** MERGE

**Ejemplo:**

## **7.15. SORTWORK - Usando una especifica clase de fila (RAP)**

`SORTWORK(numero par1)`

**Parametros:** *par1* : Numero de clasetrabajofila

**Descripcion:** Durante la clase RAPGEN creas las filas de trabajo: c:/tmp/SIN00000.000 y c:/tmp/SUD00000.000 donde c:/tmp/ es el normal TMP directorio. Estas clases de filas no estan suprimidas despues de usarlas como tu por empezar de el proximo registro por estar entrando

### **EMPEZAR A: CLASE o SORTD**

puedes evitar el tiempo de la clase y usar la misma clasificacion como para correr la pasada. Si tu intentas usar esta funcion `SORTWORK(47)` puedes evitar que la clase de las filas no estan escritos encima por otras listas como los nombres de la fila cuando empiezan: c:/tmp/SIN00000.047 y c:/tmp/SUD00000.047.

**Volvervalor:** None.

**Ver tambien:**

**Ejemplo:** `SORTWORK(47)`

## **7.16. WHEN** - Cuando realizas los calculos (RAP)

WHEN(numero *par1*, numero *par2*)

**Descripcion:** El comand WHEN es usado para definir cuando los calculos pueden ser realizados, i.e. antes/despues de estar clasificanco o acumulando totales.

## **8. Marcar control**

Esta seccion describe funciones para el registro de control marcado.

## **8.1. COPIES**- Numero de copias marcadas (RAP)

COPIES(numero *par1*, Marcar *par2*)

*par2* : numero opcional marcado

**Descripcion:** COPIES(1) coges una copia adicional de la marca de la salida. Un maximo de 30 copias puede ser empezada y deberia ser habitacion para todas las ventanas filas del carrete. COPIES(1,7) produce una copia adicional en la marca definida como no.7 en la marca setup. Anotar como siempre que inesperadamente cambio de las paginas deberian de ocurrir sif la copia marcada tiene una pequeña forma mas que el original.

**Volvervalor:** None.

**Ver tambien:** PRINT

**Ejemplo:** COPIES(1) /\* Marcar 2 tiempos

## **8.2. PAGE** - Cambiar registro de distribucion de la pagina (RAP)

numero PAGE(numero *par1*)

**Parametros:** *par1* : el registro de la pagina necesitado

**Descripcion:** Un registro normalmente usas la pagina 0 cuando estas marcando. Esta es la pagina que tu mormalmente usas cuando estas definiendo una distribucion. Un registro puede usar diferentes distribuciones de las paginas eg. permite distribuir letras que estan marcadas en diferentes lenguajes (max. 9 distribucion de las paginas). Estas paginas son numeradas de 0 a 9 y pueden ser alcanzadas de la 'fila' menu, 'distribucion de la pagina' cuando estas editando la forma.

**Volvelvalor:** Vuelves la pagina actual seleccionada como marca activa de la pagina.

**Ver tambien:** PRINT

**Ejemplo:**

### 8.3. **PRINT** - Marcar lineas de registros de la distribucion (RAP)

PRINT(texto *par1*)

**Parametros:** *par1* : las lineas son marcadas

**Descripcion:** La funcion es usada a marcar lineas de la distribucion del registro, o to set una marca en el comando que esta realizada para cada pagina o marca de las lineas totales. La sintaxis es:

Funcion	Descripcion
PRINT(1-10)	Marcar las lineas 1 a 10
PRINT(1,+2,2)	Marcar linea 1, cuando 2 lineas en blanco y finalmente la linea 2
PRINT(1,:60,2)	Marcar linea 1, goto linea 60 y marcar linea 2
PRINT(:1003,1,3)	Goto 3 lineas antes que el boton marcado y la linea 1 y 3
PRINT(1-10,:1,20)	Marcar linea 1 a 10, hacer forma alimentacion y marca de la linea 20
PRINT(*H)	Las lineas definidas usando H= esta marcada

La funcion es tambien usada to set marcar comandos controlando las lineas cuales marcas en diferentes situaciones:

Funcion	Descripcion
PRINT(H=1-4)	Cuando una nueva pagina, marcar lineas 1 a 4
PRINT(L=8)	Las lineas marcadas para cada registro en la fila principal es marcada en la linea 8
PRINT(T=10)	El total de la linea es la linea 10 (pones tambien para totales grandes)
PRINT(D=9)	Como encabezado para el detalle de las lineas(READ) linea 9 esta marcada
PRINT(B=:1002,17)	Como boton en cada pagina la linea 17 esta marcada
PRINT(N=3,:1,1-4)	Nuevapagina 3 lineas antes de acabar la pagina, encabezado de la linea 1-4
PRINT(A=10)	Linea 10 es marcada primero a un bloquetotal
PRINT(C=11)	Linea 11 si marcas despues de un bloque total

Anotar que un texto del campo puede ser usado en el comando marcado como

PRINT(>2) cambios a marcar 2, ver PRINT.

**Volvalor:** None.

**Ver tambien:** PAGE , PRINT

**Ejemplo:** PRINT(:60,1-10) /\* goto linea 60 y marcar linea 1 a 10

### 8.3.1. **PRINT** - Marcar salida de control (RAP.)

PRINT(texto *par1*)

**Parametros:** *par1* : opcion=valor

**Descripcion:** La PRINT del comando esta expandida con la sintaxi del comando PRINT(xx=valor yy), donde xx,valor y yy puede ser uno de los siguientes:

	<b>Funcion</b>	<b>Descripcion</b>
xx=	ml	Margen izquierdo
	mr	margen derecho
	mt	Top margen
	mb	Boton margen
	eh	Altura de la linea vacia
	ce	Cerrar ventanas del registro en la salida
	fh	Empezar altura de la fuente para todas las lineas
	cd	Cerrar documento marcado y empezar nuevo
yy=	cm	Centimetro
	in	Pulgadas
	pt	Puntos
	<none>	aparato pixels

### **8.3.2. PRINT(=?= - Marcar las características de la pregunta (RAP.)**

`PRINT(=?=textO par1)`

**Descripcion:** El comando PRINT esta tambien expandido con una pregunta de la funcion en orden a recibir alguna informacion del manejo interno marcado.  
El volver valor yy esta registrado en pixels excepto cuando xx es 5, 8, 9, 15 o 16.

## **8.4. PRINT(LAB= - Funcion de la etiqueta (RAP)**

PRINT(LAB=Texto *par1*, Texto *par2*, Texto *par3*, Texto *par4*, Texto *par5*, Texto *par6*)

*par6* : Copias

**Descripcion:** La longitud y altura de alguna etiqueta en el paso puede ser cogida en centimetros o pulgadas por estar usando la siguiente sintaxis:

La muestra de abajo produce etiquetas marcadas de izquierda a derecha en una etiqueta de la hoja con 21 etiquetas, 3 en cada linea, 7 lineas, donde cada etiqueta tiene la longitud/altura de 7 centimetros. Cada etiqueta es marcada en 2 copias.

**Volvervalor:** None.

**Ver tambien:** PRINT

**Ejemplo:**

## **8.5. PRINTER- Marcar seleccion (RAP.)**

PRINTER(Marcar *par1*)

**Parametros:** *par1* : Marcarnumero

**Descripcion:** Esta funcion es usada en conexion con la marca del dialogo. En orden a set la falta marcada para un registro de la linea siguiente puede ser añadida en los calculos:

**Volvervalor:** None.

**Vertambien:** COPIES, PRINT

**Ejemplo:** PRINTER(7) /\* falta en la marca para este registro que esta marcado 7

### **8.5.1. PRINTER - Multiples marcas en la salida (RAP)**

PRINTER(numero *par1*, Marcar *par2*)

**Parametros:** *par1* : Marcarnumero *par2* : MarcarID

**Descripcion:** PRINT(2,7) debe de abrir la segunda marca definida como numero marcado 7 en la marca setup. No salida esta marcada en este hasta una

**PRINT(>2)**

es encontrado en los calculos donde despues todas las marcas van a esta marca. PRINT(>1) interrptores en blanco o falta de marcas.

Cada marca tiene su propio numero de pagina y puede diferir el tamaño en el papel. Un maximo de 30 marcas concurrentes o copias pueden ser usadas.

**Volvervalor:** None.

**Vertambien:** COPIES, PRINT

**Ejemplo:** PRINTER(2,7) /\* *Abrir la segunda marca 7*

## **8.6. PRTTOTAL - Manual de control de la marca de fuera (RAP)**

PRTTOTAL(Nivel *par1*)

**Parametros:** *par1* : Nivel total del numero

**Descripcion:** RAPGEN normalmente produce un subtotal cuando una parte de la clase de llave cambia valores. Con el uso de PRTTOTAL tu puedes normalmente controlar todas las marcas de subtotales y en lugar de marcas estos cuando un campo cambia el valor.

**Volvervalor:** None.

**Ver tambien:** ENDSUM

**Ejemplo:**

## 8.7. **SCRPR** - Rellamar pantalla marcada (IQ)

SCRPR(Nombrefila *par1*)

**Parametros:** *Par1*: Nombre fila demostrado usando la marca de la pantalla

**Descripcion:** SCRPR("nombrefila") llamas encima de la marca de la pantalla con la marca de salvar de nombrefila. Esto puede por ejemplo ser usado en IQ por hacer un click en un campo.

**Volvervalor:** None.

**Ver tambien:** PRINT

**Ejemplo:**

## **9. Leyendo filas**

Este capítulo describe el READ función para estar leyendo un registro de una fila secundaria y el START/NEXT/REPEAT funciones para el lazo de encima un rango de registros.

Los principales manejos de mas filas y estas conexiones estan descritas en RAPGEN manualusuario, seccion Usando multiples filas.

## 9.1. **READ** - Leer el registro de una fila

numeor READ(fila *par1*, indice *par2*) ,conexion *par3*

*par3* : conexion opcional si empiezas la conexion no presente o convenientemente

**Descripcion:** La funcion lee un registro de una fila.

READ(le) lees la fila le usando el comienzo de la conexion definida en la Fecha Diccionario.

READ(le),#9 lees la fila le usando el campo 9 como llave para el indice 1, nomateria si y como un comienzo de la conexion esta definida.

READ(va.02),#6 lees la fila va usando el campo 6 como llave para el indice 2, nomateria si y como un comienzo de la conexion esta definida.

READ(le),"1",#9(3,4),#7 formas la llave como una combinacion de el constante "1" y caracter 3-4 de campo 9 preguntado por el campo 7.

READ(le.00),#6 llee el le de la fila usando el numero de registro (indice 0) como cogarlo en el campo 6.

**Volvervalor:** 0 si el registro esta leido.

**Ver tambien:** START, NEXT, REPEAT, END , PRIOR, READH, READX

**Ejemplo:** READ(le) /\* leer el distribuidor

## 9.2. **READH** - leer registro con marca opcional del encabezado

numero READH(fila *par1*, indice *par2*) ,conexion *par3*

*par3* : Conexion opcional si al empezar la conexion no presenta o conveniente

**Descripcion:** La funcion de leer un registro de una fila justamente como READ. Si otro registro esta leido que el pasado tiempo READH estaba usado, eg. cuando el numero del distribuidor cambia, el encabezamiento cogido para READH debe de ser marcado.

**Volvervalor:** 0 si el registro esta leido.

**ver tambien:** READ

**Ejemplo:** READH(le) /\* leer el distribuidor con el encabezamiento opcional

### **9.3. READR** - Leer registro usando el numero de registro

numero READR(fila *par1*) ,conexion *par2*

*par2* : Opcional conexion si al empezar la conexion no presenta o conveniente

**Descripcion:** Esta funcion lee un registro de una fila usando el numero de registro como llave. READR puede ser usado unico en la fechabase de los sistemas que estas trabajando con el numero de registro y estan incluidos unicamente para compatibilidad con relaciones previas.

READ(le.00),#6 es el mismo como READR(le),#6

**Ver tambien:** READ , READX

## 9.4. **READX** - Leer registro usando el numero de registro relativo

numero READX(fila *par1*) ,conexion *par2*

*par2* : Opcional conexion si al comenzar la conexion no presenta o conveniente

**Descripcion:** Esta funcion lee un registro de una fila usando el numero de registro relativo como llave. READX puede ser usado unico en la fechabase de los sistemas que estas trabajando con un numero de registro y estan incluidos unico para compatibilidad con relaciones previas.

READ(le.00),#6+N es el mismo como READX(le),#6

**Ver tambien:** READ , READR

## 9.5. **START** - Set indice y rango para una fila

numero START(fila *par1*, indice *par2*) ,conexion *par3*

*par3* : Opcional conexion si al empezar la conexion no presenta o conveniente

**Descripcion:** La funcion prepara la lectura con la PROXIMA funcion por setting el rango de las llaves para este.

El comienzo de la conexion de la fila puede ser usado o la llave puede ser especificada justamente como descrita para READ.

Por START tu deberias normalmente justamente especificar una parte de la llave. La subsecuencia esta leyendo con NEXT deberia cobrar todos los registros donde la primer parte de el registro de la llave iguales con la parte de la llave cogida al START.

**Volvervalor:** Volver 0 si el rango ok.

**Ver tambien:** READ, NEXT, REPEAT, END , PRIOR

**Ejemplo:**

## 9.6. **NEXT** - Coger proximo registro en rango

numero NEXT(fila *par1*)

**Parametros:** *par1* : nombrecorto de la fila

**Descripcion:** La funcion es usada en la conexion con START/NEXT/REPEAT lazos. Las funciones START() y END() set la respuesta del rango para el lazo. NEXT() cuando lees un registro de la fila. Cuando el calculo REPITE() se realiza la funcion NEXT() debe de ser realizado alguna vez ahora hasta no mas registros existentes en el rango cogido.

**Volvervalor:** Volver 0 como largo como registros existentes en el rango.

**ver tambien:** READ, START, REPEAT, END , PRIOR

**Ejemplo:**

## 9.7. **REPEAT** - Repetir leyendo NEXT

numero REPEAT(fila *par1*)

**Parametros:** *par1* : nombrecorto de la fila

**Descripcion:** La funcion es usada en la conexion con START/NEXT/REPEAT lazos. Las funciones START() y END() set el rango necesitado para el lazo. NEXT() cuando lees un registro de la fila. Cuando el calculo REPEAT() es necesitado la funcion NEXT() debe de ser necesitado alguna vez ahora hasta no mas registros existentes en el rango cogido.

**Volvervalor:** None

**Ver tambien:** START, NEXT , PRIOR

**Ejemplo:**

## 9.8. **GETKEY** - Coger actual valor de la llave

texto GETKEY(campo *par1*)

**Parametros:** *par1* : Campo

**Descripcion:** #20=GETKEY(va) volver el indice de la llave para la pasada lectura del registro en la fila va. La funcion es designada especialmente para la fechabase de los sistemas donde la llave no es necesariamente tiene que ser parada como un campo en la fecha del registro.

**Volvervalor:** El valor de la llave como texto.

**ver tambien:**

**Ejemplo:** #20 = GETKEY(va)

## 9.9. **END** - Set fin rango para una fila despues de START

numero END(fila *par1*) ,conexion *par2*

*par2* : fin rango especificacion

**Descripcion:** Al START la funcion define al empezar la llave y al acabar la llave igual como la primera parte completa de la llave. Por ejemplo todos los correos con igual numero debido esta leido.

Normalmente tu no tienes que usar END, esto es unicamente necesario si tu necesitas un rango final especial.

**Volvervalor:** Volver 0 si rango ok.

**Ver tambien:** READ, START, REPEAT, NEXT , PRIOR

**Ejemplo:**

## 9.10. **PRIOR** - Coger registros previos en rango

numero PRIOR(fila *par1*)

**Parametros:** *par1* : nombrecorto de la fila

**Descripcion:** PRIOR trabajo justamente NEXT pero registro previo es cobrado. Anotar que no todas las fechasbase interfaces soportan leyendo los registros en 'reversa' orden.

**Volver valor:** Volver 0 como largos como los registros existentes en el rango.

**Ver tambien:** READ, START, REPEAT, NEXT , END

**Ejemplo:**

## 9.11. **VELOCIDAD**- Optimizar lectura estrategica

VELOCIDAD()

**Parametros:** none

**Descripcion:** La VELOCIDAD() funcion puede ser usada para optimizar la lectura estrategica en un registro como un registro no debe de ser leído ahora cuando la misma llave es cogida pero coges de la memoria. Tu deberias de tener cuidado con este en la actualizacion de los registros.

**Volvervalor:** None.

**Ver tambien:** READ

**Ejemplo:** VELOCIDAD()      /\* El registro optimo leído

## 10. Escribiendo a filas

Este capítulo describe los diferentes caminos de actualizar las filas. Usar de estas funciones necesitadas que el subsistema esta instalado permitiendo actualizar de las filas, que el uso de la fecha base tiene funciones para este y que el usuario tiene escritos permisos en el servidor. Algun programa hecho en la fila actualizado deberia ser comprobado antes de usarlo. Esto deberia ser

**el total responsabilizado del usuario**

que la actualizacion de acuerdo tiene que ser comprobado y esta trabajando correctamente.

## **10.1. UPDATE - Permitir actualizar filas**

numero UPDATE(numero *par1*, campos *par2*)

*par2* : Fila opcional/permitir actualizar los campos.

**Descripcion:** UPDATE(1) debe de ser reemplazado en una actualizacion del registro antes de alguno de lo escrito en las funciones estan usadas en activar esta orden.

La actualizacion del comando tiene que ser extendida con especificacion de los campos a actualizar.

**Volvervalor:** None.

**Ver tambien:** DELETE, INSERT, REWRITE, WRITE, NOPAS

**Ejemplo:**

## **10.2. REWRITE - Reescribir registro en la fila**

numero REWRITE(fila *par1*)

**Parametros:** *par1* : nombrecorto de la fila

**Descripcion:** La funcion actualiza un registro al coger en la fila las cuales deben de estar leidas. Indice de los campos pueden ser modificados unicamente si el uso de la fechabase del sistema de estos soportes. El calculo UPDATE(1) deben de tener que ser ejecutados a activar esta funcion.

**Volvervalor:** 0 si el registro tiene que ser actualizado.

**Ver tambien:** DELETE, INSERT, WRITE, NOPAS, UPDATE

**Ejemplo:**

### **10.3. INSERT** - Insertar nuevo registro en la fila

numero `INSERT(fila par1)`

**Parametros:** *par1* : nombre corto de la fila

**Descripcion:** La funcion inserta un nuevo registro en una fila. TODOS los campos en la fila tienen que ser asignados a un valor prior a `INSERT`. La ACTUALIZACION del calculo(1) deberia de tener que ser ejecutado a esta funcion activada.

**Volvervalor:** 0 si registro esta insertado.

**Ver tambien:** DELETE, REWRITE, WRITE, NOPAS, UPDATE, CLEAR, LET

**Ejemplo:**

## 10.4. **DELETE** - Suprimir un registro en una fila

numero DELETE(fila *par1*)

**Parametros:** *par1* : el nombre corto de la fila

**Descripcion:** La funcion suprime un registro en la peticion de la fila. El registro tiene que ser leído antes de DELETE puede ser hecho. La actualizacion del calculo(1) deberia tener que ser ejecutado a activar esta funcion.

**Volvervalor:** 0 si registro esta suprimido.

**Ver tambien:** INSERT, REWRITE, WRITE, NOPAS, UPDATE

**Ejemplo:**

## **10.5. WRITE - Escribir un registro en una fila**

numero WRITE(fila *par1*)

**Parametros:** *par1* : nombre corto de la fila

**Descripcion:** La funcion de actualizar o insertar un registro en det coger la fila. Si el pasado READ en esta fila tiene que ser encontrado un registro la funcion resulta un REWRITE, si no registro deberia de ser encontrado por READ y INSERT es usada. La ACTUALIZACION del calculo(1) debe de tener que ser ejecutado a activar esta funcion.

**Volvalor:** 0 si reistro esta actualizado/insertado.

**Ver tambien:** INSERT, REWRITE, DELETE, NOPAS, UPDATE

**Ejemplo:**

## **11. Exportar / Importar de las filas externas**

Este capítulo describe las funciones para leer/escribir de los textos de las filas con fecha para transferir a otros sistemas.

## **11.1. EXPORT** - Exportar de la fecha a un texto de la fila

numero EXPORT(campos *par1*, nombrefila *par2*, texto *par3*, texto *par4\*6*, texto *par5*, texto *par6\*6*)

**Descripcion:** EXPORT exportar fecha a un texto de la fila. La funcion puede ser usada para transferir la fecha entre sistemas, hoja electronica y procesando palabra de los sistemas. Los campos que estan cogidos en *par1* tienen que ser entrados como texto, eg. "#1-99" (en cuotas).

El nombrefila en *par2* es falta a TMP si un directorio esta omitido, la falta de la extension es .OUT y si el nombre de la fila es completamente omitido el nombre del registro como c:/tmp/DM1007.OUT para el registro numero 7.

Con *par3* y *par5* tu puedes controlar la longitud del registro y la separacion de las lineas para la fila.

*par4* es normalmente usado unicamente con la longitud fijada de las filas para transportar a el ordenador central de sistemas.

*par6* consiste de 6 caracteres usados para controlar la distribucion de una comaseparada de la fila. Anotar que " en esta cuerda tu deberias de estar escribiendo como los dos caracteres \". Como standard todos los campos alfanumericos estan escritos como "xxxx", donde el caracter " (cuota) esta convertido a una ' (unica cuota). Los campos numericos estan escritos como 99.99, donde . (dot) es el punto decimal. Todos los campos estan separados con , (coma).

La fila exportado puede ser ahora cerrada usando EXPORT("CERRAR"). Esto puede ser util si tu quieres CHAIN del camino de la nota a ver en la fila.

**Volvervalor:** None.

**Ver tambien:** IMPORT

**Ejemplo:**

Calculos y subfunciones

Este ejemplo debe de ser creado en la fila le.csv con las siguientes lineas:

**Ejemplo:**

Este ejemplo debe de crear la fila le.ssv con las lineas siguientes:

**Ejemplo:**

## 11.2. **IMPORT** - Importar fecha de la fila del texto (RAP)

IMPORT(campos *par1*, nombrefila *par2*, texto *par3*, texto *par4\*6*, texto *par5*, texto *par6\*6*)

**Descripcion:** La funcion leer fecha de una fila del texto.

Los campos que estan cogidos en *par1* tienen que ser entrados en "" (cuotas). Esto es posible de coger unos calculos simples siempre con la especificacion del campo como IMPORT("#1-5,+6")

<b>Operador</b>	<b>Funcion</b>
+	Añadir encima de estos campos
-	Sacar de estos campos
&	Saltar estos campos
=	Set campos igual
:xx	Saltar a la posicion xx en el registro

El nombre fisico de la fila cogido en *par2* puede contener un camino, e.g. "c:\\export\\le.csv".

**Volvervalor:** None.

**Ver tambien:** [EXPORT](#)

**Ejemplo:**

**Ejemplo:**

### **11.2.1. IMPOCONT - Continuacion del importe (RAP)**

IMPOCONT(campos *par1*)

**Parametros:** *par1* : los campos son ser por leer de la fecha de la fila del texto

**Descripcion:** IMPOCONT continuas importe de mas campos de el mismo registro y posicion del registro como el pasado IMPORT buscado. Usando cuando un tipo de registro por empezar del registro puede causar importe de diferentes campos.

**Ver tambien:** IMPORT, IMPONEXT, IMPOESTE

## **11.2.2. IMPONEXT - Importe del proximo registro (RAP)**

IMPONEXT(campos *par1*)

**Parametros:** *par1* : los campos estan set por leer la fecha de la fila del texto

**Descripcion:** IMPONEXT lees un nuevo registro del importe de la fila y importes de los campos de aqui. Usar cuando un campo indica que uno o mas registros siguientes con relatados en las informaciones para este registroprincipal.

**Ver tambien:** IMPORT, IMPOCONT, IMPOESTE

### **11.2.3. IMPOESTE - Reimporte de este registro (RAP)**

IMPOESTE(campos *par1*)

**Parametros:** *par1* : los campos son set por leer de la fecha de la fila del texto

**Descripcion:** IMPOESTE importes del registro actual de ahora. Usado cuando un tipo de registro por empezar del registro puede causar el importe de diferentes campos.

**Ver tambien:** IMPORT, IMPOCONT, IMPONEXT

### **11.3. FTP - Procesador Transferido de la fila**

numero FTP(Numero *par1*, Texto *par2*)

*Par2*: FTP comando

**Descripcion:** La funcion FTP tienen que ser construida en permitir avanzar a los usuarios a transferir filas por ejemplo en un registro basado en un SSV fila conteniendo los nombres de la fila. Por un completo set de comandos tu deberias consultar un FTP manual. Anotar que los campos libres pueden ser usados para el comando y que los 32 bit version sujetan nombre largos de la fila.

El ejemplo demostrado la transferencia de un fila de un Quattro sistema con el comando especial QUATTRO para transferir con el bloque de cabeza y XQUAT mover el adicional FTP informaciones de el tambien transferida la fila.

**Volvalor:** para OPEN: FTP Manejo, todos los otros: FTP error codigo, 0=OK

**Ejemplo:**

## **12. Múltiples compañías y unión de las filas**

Estas funciones describen debajo están intentando usar con múltiples compañías con tablas separadas/fechabase de los sistemas y para estar uniendo diferentes filas con el mismo registro de la distribución.

## 12.1. **ACCESS**- Controlar si fila existe (IQ)

numero ACCESS(Nombrefila *par1*)

**Parametros:** *par1* : Nombrefila

**Descripcion:** Controlar si la fila cogida es presente, volver 0 si la fila esta encontrada.

**Volvervalor:** 0 si fila es encontrada.

**Ver tambien:** OPEN

**Ejemplo:** IF ACCEDES("myfile.ssv")=0 MESS("Ok ? ")

## 12.2. **COMNO** - Coger compañía actual id

texto COMNO(Filaid *par1*)

**Parametros:** *par1* : Espacio de filaid

**Descripcion:** Esta funcion conbra la actual compañía id de la fila cogida, si no fila esta cogida para la fila principal.

**Volvervalor:** Compañia id.

**ver tambien:** OPCOM

**Ejemplo:** #1 = OPCOM()      /\* Coger compañía actual id, eg. "001" \*/

## **12.3. ENDSUM** - Total grande adiciona cuando estan usando mas filasprincipales

ENDSUM()

**Parametros:** None

**Descripcion:** En un registro con mas listas separadas causadas por el uso de otros la MERGE o el OPCOM funcion con los totales para cada lista tu puedes obtener un total adicional de todas las marcas de los registros por reemplazar un ENDSUM() calculo de la linea.

Los campos del sistema #CO y #CN deben de ser marcados como \*\*\* a el ENDSUM pagina.

**Volvervalor:** None

**Ver tambien:** KEYS, MERGE, OPCOM

**Ejemplo:** ENDSUM() /\* marcar fin en el total adicional

## **12.4. FILENAME** - Actual nombrefila para una fila abierta

texto FILENAME(filaid *par1*)

**Parametros:** *par1* : Filaid

**Descripcion:** Esta funcion vuelve al nombre de la fila para la fila las cuales estan actualmente abiertas con la filaid cogidas.

**volvvalor:** Nombre real de la fila.

**Ver tambien:** OPEN

**Ejemplo:** #1 = FILENAME(va)     /\* Coges "c:/rapfil/ssv/isa/va.ssv"

## **12.5. OPEN** - Abrir una fila con un nombre específico

numero OPEN(filaid *par1*, Nombrefila *par2*, Conductor *par3*)

*par3* : 0 o fechabase interface numero

**Descripcion:** Con el uso de esta funcion tu puedes abrir una fila especifica dentro de un deacuerdo abirtas y asociadas con este campoid. La anterior fila abierta deberia de estar cerrada.

Un error mensaje esta cogido si el nombre de la fila no esta presente o la fila no puede ser abierta para alguna otra razon.

si *par3* esta cogiendo la fila abierta esta forzada con esta fechabase interface tipo como definido en BASIS.SSV por el conductor instalado fechabase.

**Volvervalor:** 0=ok, <>0=error.

**Ver tambien:** ACCESS, FILENAME, MERGE, OPCOM

**Ejemplo:**

## 12.5.1. **OPEN** - Temporamente cerrar filas

OPEN(filaid *par1*, Constante *par2*)

*par2* : "-"

**Descripcion:** Las filas pueden ser cerradas permitiendo ENCADENAR programas a acceder esta NOTA que la fila PRINCIPAL no puede ser cerrada en este camino.

**Volvervalor:** 0=ok, <>0=error.

**Ver tambien:** FILENAME, MERGE, OPCOM

**Ejemplo:**

## **12.6. MERGE** - Uniendo mas de dos filas principales en un registro (RAP)

numero MERGE(filaid *par1*, Nombrefila *par2*, Conductor *par3*)

*par3* : 0 o numero de la fechabase interface

**Descripcion:** Con el uso e esta funcion tu puedes unir multiples filas dentro de una lista. Para la MERGE de una rutina tu puedes coger cada filaid en *par1* si la fila esta definida separadamente o el nombrefila en *par2* como en OPEN. Las filas complicadas deberian de tener la misma estructura.

Un error en el mensaje es cogido si el nombre de la fila no esta presente o la fila nopuede ser abierto para alguna otra razon.

Si *par3* esta cogiendo la fila es forzada a abrir con esta fechabase interface tipo como definir en BASIS.SSV por el conductor de la instalacion fechabase.

Un registro esta usando MERGE deberia normalmente ser clasificar el aumento de la union efectiva, eg. clasificar en el numero de articulo. Si OPEN es usar sin estar clasificando tu deberias coger la primera lista una lista de el normal fila principal preguntando por una lista de cada una de la union de las filas. El ENDSUM funcion puede ser usada para coger un total grande de todos los registros marcados.

Si OPEN es llamado sin parametros a todos un NUMEROUNIDO es devuelto como 1 para la fila principal, 2 para la primera fila unida, 3 para el proximo y tambien en.

Sin parametros: NUMEROUNIDO de 1 y adelante.

**ver tambien:** ENDSUM, OPCOM, OPEN

**Ejemplo:**

## 12.7. **OPCOM** - Abrir filas en diferentes compañías

numero OPCOM()

*par3* : 0 o numero de la fechabase interface

**Descripcion:** El OPCOM funcion permite que tu accedas a mas compañías en un registro.

Un registro puede ser hecho para correr alguno para cada compañía al empezar por reemplazar OPCOM("111,777-888") o OPCOM(#50) donde el campo 50 es empezar una fecha en la entrada del campo. Un registro parecido puede ser extendido cuando con un total para todas las compañías usando ENDSUM o puede ser clasificado eg. coleccionar todas las informaciones de un articulo en todas las compañías.

El sistema de los campos #CO y #CN puede ser usada para marcar la compañía id y el nombre de la compañía en el encabezamiento.

La lista de un articulo definida en la fila va puede ser hecha a coleccionar informaciones del articulo de otra compañía tambien reemplazada OPCOM(VA,"555") preguntando por READ(VA). Por esto va#8 contiene la propiedad para la actual compañía donde como VA#8 es la propiedad para la compañía 555.

Un registro estatico donde cada registro en las estadisticas de la fila contiene un numero de compañía puede abrir estas filas de la compañía por usar de OPCOM(0,#47)

Si *par3* es cogida la fila es abierta forzada con esta fechabase interface tipo como definida en BASIS.SSV por fechabase del conductor instalado.

Si OPCOM es usado sin parametros el numero de la compañía actual es devuelto.

Para la COMPANYY.SSV fila los nombres de la compañía son leidos. Si la compañía id's en *par3* contiene rangos las compañías validas de aqui estan cogidas para esta fila.

Sin parametros: UNIRNUMERO de 1 y adelante.

**ver tambien:** COMNO, ENDSUM, MERGE, OPEN

**Ejemplo:**

## **13. IQ/DATAMASTER funciones**

Las funciones estan designadas para DATAMASTER y no puede ser usada en registros. Algunas de estas funciones estan provechoso tambien en IQ los cuales pueden ser indicadas cuando en el texto.

### **13.1. DISABLE**- Estropear la entra para un programa (IQ)

DISABLE(programano *par1*)

**Parametros:** *par1* : Numero de program estropeado.

**Descripcion:** Estropear todas las entradas para coger el numero de programa.

**Volvervalor:** None.

**Ver tambien:** ENABLE , FOCUS

**Ejemplo:** DISABLE(20)

## 13.2. **DISP** - Display de cambiar los campos (IQ)

DISP(campos *par1*)

**Parametros:** *par1* : "" o campos a redisplay

**Descripcion:** DISP es ser usado cuando tu en un calculo para los calculos cambiados el valor de otros campos y estos campos estan demostrados en la pantalla. Si DISP no es presente y tu nopuedes ser cierto lo nuevamente calculado en el valor de acuerdo esta demostrado.

El DISP() comando displaying todos los campos estan extendidos posiblemente de estado justamente seleccionado en los campos como DISP("#1,4")

**Volvervalor:** None.

**Ver tambien:**

**Ejemplo:** DISP()

### **13.3. DOFUNCTION - Ejecutar funcion externa (IQ)**

DOFUNCTION(Funcion *par1*, texto *par2*, programano *par3*)

*par3* : Programano opcional

**Descripcion:** DOFUNCTION enviar el mensaje <funcionno> a el estar corriendo IQ-Programa o al abrir <programa>. Una llave puede ser pasada a la READ funciones.

La lista de la funciones de los numeros validos estan encontradas en los calculos de la caja de la lista para 'Calculos por seleccion de la funcion'.

**Vovlervvalor:** None.

**Ver tambien:** CHAIN, PLSNEXT, TRANSMIT

**Ejemplo:**

### **13.4. ENABLE**- Permitir la entrada para un programa (IQ)

ENABLE(Programano *par1*)

**Parametros:** *par1* : Programa numero permitido

**Descripcion:** Permitir todas las entradas para los otros numeros del programa cogidos despues ESTROPEADA.

**Volvervalor:** None.

**ver tambien** DISABLE , FOCUS

**Ejemplo:** ENABLE(20) /\* Permitir programa 20

### **13.5. FOCUS - Activar programa (IQ)**

FOCUS(Programano *par1*)

**Parametros:** *par1* : Programanumero activado.

**Descripcion:** Activas la entrada y sets focos a coger el numero del programa.

**Volvervalor:** None.

**Ver tambien:** DISABLE, ENABLE

**Ejemplo:** FOCUS(20) /\* Programa 20 esta activado

## **13.6. FUNC** - Actualizacion del modo actual para un registro (IQ)

numero FUNC(filaid *par1*)

**Parametros:** *par1* : Filaid

**Descripcion:** Depende de los usuarios de la entrada en DATAMASTER decides si la actualizacion de un registro cierto es necesariamente y como este deberia ser realizado. FUNC es cuando usado en la escritura de los calculos a rama a la proxima rutina.

**Volvervalor:**

<b>Modo</b>	<b>Funcion</b>
0	No actualizar necesariamente
1	Un registro existente deberia de ser modificado
2	Un nuevo registro deberia ser insertado
3	Un registro existente deberia de ser suprimido

**Ver tambien:** SETUPD, ON

**Ejemplo:**

## **13.7. GETINFO - Coger programa adicional de informacion (IQ/DM)**

numero GETINFO(numero *par1*, texto *par2*)

*par2* : Campo de referencia

**Descripcion:** Estas funciones permiten que tu cojas la misma informacion especial de un IQ/DM programa. El tipo 0 y 1 debe de volver el unico id de la ventana, los cuales pueden ser usados por otras funciones a manipular la ventana. Una misma de estos es presente en el OLE manual.

Cuando el tipo es 2 a 5 la funcion necesita una referencia en el campo en *par2*. Por ejemplo, coger al empezar la columna para el numero de articulo 7 *par2* deberia de ser igual "va#7". Los coordinados de un campo estan aqui cogidos en el actual tamaño del campo definido en IQ/DM.

Si tu necesitas los coordinadores actuales de un campo acordando a la escala factor actualmente usado, e.g. zoom en/fuera, usar el tipo 6 a 9 instalado.

Tipo 2-9 volver al campo coordinado. El valor puede ser agarrado en un 9,T2 formato del campo.

**Ejemplo:**

## **13.8. HELP** - Display caja con ayuda para campo (IQ)

HELP(campo *par1*)

**Parametros:** *par1* : referencia del campo

**Descripcion:** HELP("#31") displays un mensaje a la caja con ayuda para coger el campo

**Ver tambien:** MESS

**Ejemplo:** HELP("#31")

### **13.9. ISACTIVE** - Preguntar si el programa esta activado (IQ)

numero ISACTIVE(Programano *par1*)

**Parametros:** *par1* : Programano

**Descripcion:** Test si <programa> esta activado.

**Volervvalor:** Volver 1 si <programa> es activado, 0 tambien.

**Ver tambien:** CHAIN, EXIT, WAIT

**Ejemplo:** IF ISACTIVE(20)=0 CHAIN(20) /\* Empezar programa 20 si no esta hecho

### **13.10. KEYON** - Interruptor de la llave de entrada en el campo ON/OFF (IQ)

KEYON(numero *par1*)

**Descripcion:** KEYON(0) mueves la llave en la entrada del campo, (1) reactivas esto.

**Volvervalor:** None.

**Ver tambien:**

**Ejemplo:** KEYON(0) */\* Mover la llave de entrada en el campo*

### **13.11. LINE** - Cobrar o set el numero de linea actual (IQ/DM)

numero LINE(numero *par1*)

**Parametros:** *par1* : Tipo de informacion coger

**Descripcion:** la funcion debe de cobrar o ser el IQ/DM numero de linea. El numero de la linea es la cuenta para las lineas definidas en un programa definido como **va#1-6l** o **le#1-6/va#1-6**.

Si *par1* iguales 0 la funcion de volver el numero de la linea de la actualmente linea activada.

si *par1* iguales -1 la funcion vuelve a el numero de las lineas definidas para los programas. Si los programas estaban definidas como **va#1-6l,t5** el valor devuelto deberia de ser **5**

si *par1* es mas grato que 0 la funcion sets la linea activa a *par1*.

**Vovlervvalor:** Un numero de la linea/cuenta o zero si las funciones sets el numero de la linea.

**Ejemplo:**

## **13.12. LOOP** - Llamar una rutina para todos los registros en la memoria de la linea (IQ)

LOOP(etiqueta *par1*)

**Parametros:** *par1* : Etiqueta (el nombre de la rutina) es llamado

**Descripcion:** para cada registro leido en una lista del programa y para cada transaccion en una transaccion del programa interno de la memoria de la linea es llenado con la lectura de los valores del campo siempre con el resultado de los calculos para que la linea (non-global campos de trabajo).

En la rutinaescrita de el tipo de programa LOOP es usado a llamar a una rutinaescrita para cada linea unica. Tambien LOOP es usado a recalcular SUM de todas las transacciones de las lineas.

**Volvervalor:** None.

**Ver tambien:** GOSUB, ON

**Ejemplo:**

### **13.13. MENUCH - Flip menu parado flag (IQ)**

MENUCH(Menuno *par1*)

**Parametros:** *par1* : Menunumeros

**Descripcion:** Flip paradaflag en los numeros del menu cogidos (ver MENUS) y actualizar la flag interna acordando para el control del programa.

**Volvervalor:** None.

**Ver tambien:** MENUUPD, MENUS

**Ejemplo:** MENUCH("31-32") /\* *Flip hablar y escuchar menu*

## 13.14. MENUS - Menu control (IQ)

MENUS(Menu no *par1*)

**Parametros:** *par1* : -xxx=Deactivate, +xxx=Activar los puntos del menu xxx

<b>Menunumero</b>	<b>Funcion</b>
1/11	Insertar nuevo registro en filaprincipal/transacciones
2/12	Modificar un registro en la fila principal/transacciones
3/13	Suprimir un registro en la fila principal/transacciones
4/14	Superindice buscado en la fila principal/transacciones
5/15	Selecciones en la fila principal/transacciones
6/16	Superindice en la definicion del campo de la fila principal/transacciones
20	Buscar, lista de entrada
21/22/23/24/25	Transacciones, Proximo/Previo/Primero/pasado/Direccion
26	Display llave durante la busqueda
27	Caja sensible buscada
31/32	Hablar/Escuchar a otros programas
41/42/43/44	Filaprincipal, Proxima/Previa/Primera/pasada
51/52/53	Calculos/Modificar forma/Salvar programa
54/55	Parametro menus
61/62/63/64	Nuevo programa, Suprimir programa, Marcar programa, Empezar programa
100-149	Indice localizado y numero de indice
999	cada cosa activada

**Descripcion:** El MENUS funcion puede ser usado a ambos en DATAMASTER y IQ a deactivar de acuerdo a los puntos del menu.

MENUS pueden tambien ser activados todos por empezar llamando IQ de ventanas con el -m+xxx o -m-xxx parametro. Especialmente modificar los calculos para un programa con los calculos desactivados tu deberias de tener seleccionados IQ como eg: C:\SWTOOLS\IQWIN -m999

**Volvervalor:** None.

**Ver tambien:** MENUCH, MENUUPD

**Ejemplo:** MENUS("-51-55") /\* Deactivar modificaciones de este programa

### **13.15. MENUUPD - Añadir/Control menu (IQ)**

MENUUPD(Menuno *par1*, numero *par2*, numero *par3*)

*par3* : Texto

**Descripcion:** Añadir / Control menu manualmente.

MENUUPD(1,2000,"My &OMenu propio") Añadir funcion 2000 un numero del menu 1.

Por seleccionar de este nuevo menupunto los calculos de las etiquetas del usuario FU2000: en la funcion de la seleccion deberia de ser realizado.

**Volvervalor:** None.

**Ver tambien:** MENUCH, MENUS

**Ejemplo:** MENUUPD(1,2000,"Mi &Menu propio") /\* Añadir funcion 2000 a el numero del menu 1.

## **13.16. NEXTFLD - Saltar a la entrada del campo (IQ)**

NEXTFLD(campo *par1*)

**Parametros:** *par1* : Numero de campo para la proxima entrada

**Descripcion:** NEXTFLD puede ser usado a escribir encima de la entrada de la secuencia fijada dependiendo en los calculos.

Siempre con las especificaciones del campo tu puedes coger el numero de programa o numero de la linea.

**Volvervalor:** None.

**Ver tambien:** NEXTFLDSEQ, SEQ

**Ejemplo:**

### **13.17. NEXTFLDSEQ** - Saltar la entrada de un campo en la secuencia (IQ)

NEXTFLDSEQ(numero *par1*, numero *par2*)

*par2* : Campo numero

**Descripcion:** Saltar a un distinto campo en una de las secuencias de los campos.

**Volvervalor:** None.

**Ver tambien:** SEQ , NEXTFLD

**Ejemplo:** NEXTFLDSEQ(2,1) /\* Salto a el primer campo cogido en la secuencia de la entrada  
2

## 13.18. OBJECTADDSTRING - Añadir cuerda al objeto (IQ)

OBJECTADDSTRING(campos *par1*, texto *par2*, texto *par3*)

*par3* : Texto a usar como indice

**Descripcion:** La funcion inserta un texto en un objeto. La funcion resulta varios dependiendo en el tipo de objeto. En orden a usar la funcion correctamente, por favor tener las siguientes reglas en memoria:

<b>Objeto</b>	<b>Significado</b>
BOTON	La funcion sets el texto displayed para el boton
COMBOBOX	La funcion de añadir a un nuevo elemento a la lista
EDITARCAJA	La funcion de sacar en texto en la edicion de la caja. Si el flag para multiples editar lineas tienen que ser set al texto deberia de ser añadido a el texto previo
LISTACAJA	La funcion añadir a un nuevo elemento a la lista

Parametro *par3* es el uso unico si el tipo de objeto es COMBOBOX o LISTBOX. El parametro debe de contener el valor normal del campo.

**Volvervalor:** None.

**ver tambien:** OBJECTCLEAR

**Ejemplo:** OBJECTADDSTRING("va#7",gr#2,gr#1) /\* Display nombre y no de usuario. como indice

## **13.19. OBJECTCLEAR** - Limpiar contenidos del objeto (IQ)

OBJECTCLEAR(campos *par1*)

**Parametros:** *par1* : Campo en la forma, e.g. va#7

**Descripcion:** La funcion limpia los contenidos de un objeto.

**volvvalor:** None.

**Ver tambien:** OBJECTADDSTRING

**Ejemplo:**

## **13.20. OBJECTGETSTRING**- Coger indice de un objeto seleccionado del articulo(IQ/DM)

texto OBJECTGETSTRING(campo *par1*)

**Parametros:** *par1* : Campo en la forma, e.g. va#7

**Descripcion:** La funcion cobra el valor normal del combocaja/listacaja del campo. Ello devuelve el valor igual a el *par3* usar cuando estamos llamando OBJECTADDSTRING.

El uso de esta funcion deberia normalmente ser por hacer un click en el combo/listacaja del campo.

**Volvervalor:** El normal (indice) valor de el actual articulo seleccionado.

**ver tambien:** OBJECTADDSTRING

**Ejemplo:**

### **13.21. PLSNEXT - Preparar y leer fila principal (IQ)**

PLSNEXT(numero *par1*, texto *par2*, numero *par3*, )

**Descripcion:** Preparar y realizar lectura de la fila principal de acuerdo a el modo cogido. Uso por los menus y por la pagina de abajo/arriba etc. Si entradaflag es set, la llave es usada, otravision leida es proxima/prior/directo.

**Volvervalor:** None.

**Ver tambien:** DOFUNCTION, TRANSMIT

**Ejemplo:** PLSNEXT(0,#1,1) /\* leer el proximo registro que esta usando #1 como llave

## **13.22. SEQ** - Cambiar la entrada de la secuencia (IQ)

SEQ(numero *par1*, campos *par2*)

*par2* : Numeros de campo en la nueva secuencia

**Descripcion:** El parametro de la pagina de informaciones de la secuencia del campo esta escrita encima por uso de esta funcion.

**Volvervalor:** None.

**Ver tambien:** NEXTFLD, NEXTFLDSEQ

**Ejemplo:**

### **13.23. SETUPD** - Marca una fila en una linea para actualizacion (IQ)

SETUPD(filaid *par1*)

**Parametros:** *par1* : Filaid para actualizar

**Descripcion:** Cuando 'criticos' campos en la fila principal estan cambiados este camino puede causar cambios de todas las transacciones de los registros. Normalmente modifिकास transacciones unicas que deberian de ser escritas.

**volvvalor:** None.

**Ver tambien:** LOOP

**Ejemplo:** SETUPD(va)

### **13.24. SHOW**- Permitir/estropear/Demostrar/Esconder un campo (IQ/DM)

numero SHOW(campo *par1*, numero *par2*)

3 = Esconder campo

**Descripcion:** Esta funcion permite que tu permitas/estropees un campo o una demostracion/esconder un campo.

**Volvalor:** None.

**Ejemplo:**

## **13.25. SUPER** - Preparar superindice buscado (IQ)

SUPER(filaid *par1*) , texto *par2*

*par2* : Llave

**Descripcion:** La SUPER funcion inicialmente el PROXIMA lectura para el uso del superindice

**Volvervalor:** None.

**ver tambien:** NEXT, START

**Ejemplo:**

### **13.26. TRANSMIT**- Actualizar otro IQ programas (IQ)

TRANSMIT( numero *par1*, texto *par2*, texto *par3*)

*par3* : Conexion opcional

**Descripcion:** Transmitir los registros actuales a uno o mas programas usando las conexiones automaticas o si esstan cogidas las conexiones estada.

**Volvervalor:** None.

**Ver tambien:** PLSNEXT, DOFUNCTION

**Ejemplo:** TRANSMIT (0,"","") /\* Actualizar todos los otros programas usando auto conexiones

### **13.27. TRANSSEL**- Definir IQ transaccion selecciones (IQ)

TRANSEL(texto *par1*, numero *par2*)

**Descripcion:** Scan la entrada cogida si alguna y definir la transaccion de las selecciones si la entrada contiene formulas como #15>0. Uso por flechas en la llave del campo **Volver valor:** None.

**Ver tambien:**

**Ejemplo:** TRANSSEL("#15>20",1)      /\* Definir seleccion

## **14. SYSTEM funciones**

Estas funciones estan designadas para usar unos especiales programas donde tu por ejemplo necesitas directamente acceder a las filas / caminos.

## 14.1. **DEBUG**- Encender debug ventana (IQ)

DEBUG(numero *par1*)

**Descripcion:** DEBUG(1) deberias de abrir una ventana las cuales las listas de todas las expresiones calculadas y otros numeros del programa/etiqueta cuando estan llevadas fuera. El DEBUG ventana esta cerrada cuando IQ esta cerrada.

**Volvervalor:** None

**Ver tambien:** WIE, WIFS

**Ejemplo:** DEBUG(1) /\* Encender debug ventana

## 14.2. **EXEC**- Ejecutar texto como calculo de la linea

EXEC(texto *par1*, Programano *par2*)

*par2* : **(IQ/DM)**programa numero

**Descripcion:**

ejecutas la cuerda del texto clasificado en el campo 20 como un calculo.

Cuando estas usando unos campos libres en el EXEC funcion tu debes de usar el WW#nn referencias las cuales tu puedes obtener de una marca del programa definido.

En general la cuerda pasada a el EXEC funcion no esta pretraducida y bloqueada como unas lineas normales de los calculos. Esto tiene especialmente importancia cuando usas en RAPGEN donde el C-Sintaxis de los calculos deberian de ser preguntadas. Nosostros fuertemente avisamos non-programas coger el uso de EXEC en RAPGEN simple sin olvidar de las funciones llamadas. Funcion invalida de los parametros que pueden delante proteger generales fallos.

Un punto deberia ser especialmente notificado para RAPGEN: #15=2 sets campo 15 igual a 2 TAMBIEN cuando usas como IF #15=2 LET #16=3. Tu deberias el doble el igual signo en la clase un estamento siguiente al C-Sintaxi cogiendo: IF (#15==2) LET #16=3

IQ: EXEC(#20,15) enciendes el programa activo 15 y ejecutas los calculos cogidos.

**Volvervalor:** None

**Ver tambien:**

**Ejemplo:** EXEC(#20) /\* Ejecutar un calculo entrado para empezar el registro

### **14.3. COGERFLD- Set SY estructura puntos (IQ)**

COGERFLD(texto *par1*)

**Parametros:** *par1* : Especificacion del campo

**Descripcion:** Esta funcion sets sistemas variables (SY#..) puntar a la definicion de el campo cogido. La definicion del campo puede ser leida cuando/cambias. Especial y programadores de uso unico.

**Vovlervolor:** None

## **14.4. INSTALL**- Activacion de las funciones externas

INSTALL(texto *par1*, texto *par2*, texto *par3*, texto *par4*)

*par4* : Mi funcion opcional del nombre

**Descripcion:** Programadores conociendo la funcion definidas de otros DLL's puede ahora incluir esto como IQ funciones.

**ANOTAR: Uso impropio de esta funcion que puede causar sistema rota de encima.**

**Volvervalor:** None

**Ver tambien:**

**Ejemplo:**

## **14.5. SYSPAR - Coger sistemaparametro**

texto SYSPAR(numero *par1*)

**Descripcion:** SYSPAR lees el sistema de parametro cogido. Unica permite mencionar los valores utiles.

**Vovlervvalor:** El sistemaparametro.

**ver tambien:** SYSPARSET

**Ejemplo:** #1 = SYSPAR(4)      /\* Coger el actual TMP camino

## **14.6. SYSPARSE** - Set valor del sistema parametro

SYSPARSE(numero *par1*, texto *par2*)

*par2* : Nuevo valor de este sistema de los parametros

**Descripcion:** SYSPARSE cambios el valor para coger el sistema de los parametros.

**Volvervalor:** None.

**Ver tambien:** SYSPAR

**Ejemplo:** SYSPARSE(4,"c:/mytmp/") /\* Set un nuevo TMP directorio

## **14.7. USERINFO** - Coger informacion sobre el usuario

texto USERINFO(numero *par1*)

17=Usuario definido

**Descripcion:** Esta funcion cogida la informacion necesaria del usuario.

El numero en *Par1* refiere a el numero del campo en la fila del sistema US donde tu puedes definir del campo 11 a 17 individualmente para cada instalacion, justamente estar con cuidado si mas tarde actualizas la version de TRIO.

**Volvervalor:** Cuerda contenida a la informacion del usuario.

**Ejemplo:**

## 14.8. WIF - Testmarcar (IQ)

WIF(texto *par1*)

**Parametros:** *par1* : Texto marcar

**Descripcion:** WIF coges el testmarcado sin distribucion de la distribucion de la pantalla a la fila c:/wif

**Volvervalor:** None

**Ver tambien:** WIFS, DEBUG

**Ejemplo:** WIF("Yo estoy aqui") /\* Fuera texto

## 14.9. **WIF**- Testmarcar (RAP)

WIF(texto *par1* , texto *par2*)

. **Descripcion:** WIF coges testmarcado a la fila c:/wif

**Volvervalor:** None

**Ver tambien:** WIFS, DEBUG

**Ejemplo:** WIF("Campo iguales %s.",#2) /\* *Testmarcado*

## **14.10. WIFS**- Testmarcado de los campos (IQ)

WIFS(campos *par1*)

**Parametros:** *par1* : Marcar campos

**Descripcion:** WIFS coges testmarcado de los valores del campo cogidos a la fila c:/wif

**Volvervalor:** None

**Ver tambien:** WIE, DEBUG

**Ejemplo:** WIFS("va#1-3,le#2") /\* Entrada de los valores del campo

## Index

### A

ABS .....32

### B

BASIC ..... 3;58

### C

CCODE ..... 60;63;64

CHAIN

...95;96;97;98;99;101;102;145;164;170

CHEX .....61;62

CLEAR..... 84;87;94;141

COLOR .....89;90

COMNO ..... 154;160

COMPANY..... 160

CONV..... 44;48;53;57

### D

DATAMASTER28;53;60;95;96;161;167;175

DATECALC

.....67;68;69;70;71;72;75;76;78;80;82

DELETE .....139;140;141;142;143

DISP ..... 163

### E

ENDSUM.....103;124;155;159;160

EXIT ..... 96;97;98;99;101;102;107;170

EXPORT..... 145;147

### F

FILENAME.....156;157;158

FIND .....46

FNA67;68;69;70;71;72;73;75;76;78;80;82

FNB ..... 67;68;69;70;72;75;76;78;80;82

FND

.67;68;69;70;71;72;74;75;76;77;78;80;

82

FNE .....72

FNF.....73

FNH ..... 33;34;35;39

FNO ..... 71;74;77

FNR ..... 33;34;35;39;40

FNU ..... 67;68;69;70;71;75;76;78;80;82

FNV67;68;69;70;71;72;75;76;78;80;81;82

FNY ..... 67;71;74;77

FRA .....35;36

FUNC ..... 167

### G

GETKEY ..... 134

### I

IMPOCONT .....148;149;150

IMPONEXT .....148;149;150

IMPORT .....145;147;148;149;150

INSERT ..... 86;139;140;141;142;143

INT ..... 36;39;41

### IQ

28;85;88;91;92;93;95;96;98;99;102;12

5;153;161;162;163;164;165;166;167;16

8;169;170;171;172;173;174;175;176;17

7;178;179;180;181;182;183;184;185;18

6;187;188;190;191;192;193;197;199

### K

KEYS.....103;104;155

### L

LEN .....47;55

LOOP ..... 173;184

LOWER..... 44;48;53;54;57

LTOT ..... 105;106

### M

MENUS.....174;175;176

MERGE ..... 112;155;157;158;159;160

MESS .....95;101;102;107;153;169

MTOT ..... 105;106

### N

NEXT7;126;127;131;132;133;135;136;186

NEXTFLD .....177;178;183

NOPAS ..... 108;109;139;140;141;142;143

NUMBER..... 18;27;45;50;51

NUMS ..... 18;50;51

### O

OCR..... 61

OPCOM..... 154;155;157;158;159;160

OPEN ..... 151;153;156;157;158;159;160

### P

PACK .....52;56

PAGE ..... 117;118

PAS ..... 108;109

POW .....38;42

PRINT

116;117;118;119;120;121;122;123;125

PRIOR ..... 127;131;132;133;135;136

PRTTOTAL ..... 124

### R

RAPDAY..... 82

RAPGEN

3;22;26;28;29;34;95;100;113;124;126;

191

READH ..... 127;128

READR ..... 129;130

READX .....127;129;130

REPEAT .. 7;126;127;131;132;133;135;136

RETURN ..... 3;13;95;111

REWRITE..... 139;140;141;142;143

RUN ..... 33;34;35;39

RUND..... 34;40

**S**

SEQ.....177;178;183  
SETUPD..... 167;184  
SGN..... 32;37;41  
SMAA.....44;48;49;53;54;57  
SOGE.....49;54  
SORTD ..... 113  
SORTKEY..... 112  
SORTWORK ..... 113  
SPOFF.....47;55  
SQR.....38;42  
SYSPAR..... 194;195  
SYSPARSET ..... 194;195

**U**

UNPACK .....52;56

UPDATE..... 86;108;139;140;141;142;143  
UPPER..... 44;48;53;54;57  
USING..... 18;45;50;58

**V**

VALCH ..... 60;63;64  
VALID ..... 60;63;64

**W**

WORDS .....49;53  
WORKD  
.....67;68;69;70;71;72;75;76;78;80;82  
WRITE..... 139;140;141;142;143

**Z**

ZERO ..... 84;87;94