

EVALUADOR DE FORMULAS

REFERENCIA DEL LENGUAJE

Distrito-K, 2006

1. Tipos de datos

Nada

Tipo nulo

Número

Un número de coma flotante de doble precisión.

Ejemplo: 12777,45 , 0,000056 , 1E18

Texto

Una cadena de texto de cualquier longitud, delimitada entre comillas dobles o simples.

Ejemplo: "hola" , 'sin comentarios'

Lógico

Los valores 'cierto' y 'falso'.

Ejemplo: cierto , falso , a < b

Bloque

Un bloque de código, que especifica una secuencia de instrucciones.

Ejemplo: {a=1;b=2;c=3}

Variable

Una variable, de cualquiera de los tipos indicados.

Ejemplo: &a (siendo a una variable)

Lista

Una lista (array) de elementos, de cualquiera de los tipos indicados.

Ejemplo: #("uno"; "dos"; "tres"; 4; 5)

Tipo

Un tipo de dato, cualquiera de los tipos indicados.

Ejemplo: tipo("uno")

2. Operadores de cálculo

+ (más)

- Suma de dos números
Ejemplo: $3 + 4$ devuelve 7
- Concatenación de dos textos
Ejemplo: "uno" + "dos" devuelve "unodos"

- (menos)

- Resta de dos números
Ejemplo: $3 - 4$ devuelve -1
- Cambio de signo de un número
Ejemplo: $-(3 + 4)$ devuelve -7

* (asterisco)

- Multiplicación de dos números
Ejemplo: $3 * 4$ devuelve 12

/ (barra de división)

- División de dos números
Ejemplo: $3 / 4$ devuelve 0,75

modulo

- Resto de la división entera de dos números
Ejemplo 7 modulo 4 devuelve 3

no

- negación de un valor lógico
Ejemplo: `no (3 > 4)` devuelve cierto

y

- Combina dos valores lógicos con el operador lógico AND
Ejemplo: `(1 > 2) y (3 < 4)` devuelve falso

o

- Combina dos valores lógicos con el operador lógico OR
Ejemplo: `(1 > 2) o (3 < 4)` devuelve cierto

3. Operador de asignación

= (igual)

- Asigna un valor (y un tipo) a una variable, definiéndola si no existe
Ejemplo: `a = 7`
Nota: No utilizar este operador para comparar dos elementos (un error bastante frecuente)

4. Operadores de comparación

== (doble igual)

Devuelve cierto si los dos elementos comparados son iguales. Es aplicable a todos los tipos, pero si los elementos son de diferente tipo, se produce un "error de tipo".

Ejemplo: 3 == 4 devuelve falso

<> (menor-mayor)

Devuelve cierto si los dos elementos comparados son diferentes. Es aplicable a todos los tipos, pero si los elementos son de diferente tipo, se produce un "error de tipo".

Ejemplo: 3 <> 4 devuelve cierto

> , >= , < , <= (mayor, mayor-igual, menor, menor-igual)

- Si los operandos son números, evalúa las comparaciones según el valor numérico de los operandos

Ejemplo: 3 > 4 devuelve falso

- Si los operandos son textos, evalúa las comparaciones según el orden alfabético de los operandos, sin tener en cuenta las mayúsculas / minúsculas

Ejemplo: "abeja" < "mosca" devuelve cierto

- Si los operandos son lógicos, se considera que cierto es mayor a falso

Ejemplo: cierto > falso devuelve cierto

- Si los operandos son de diferente tipo entre sí, o de cualquier otro tipo que los arriba indicados, se produce un "error de tipo"

5. Otros operadores

() (paréntesis)

- Agrupan operaciones, alterando su precedencia. Se permiten ilimitados niveles
Ejemplo: $(a + b) * (c + (d - e) / 2)$
- Indican los argumentos o parámetros de las funciones
Ejemplo: EnLetras(a)
- Precedidos por el símbolo #, definen una lista de elementos
Ejemplo: #("uno";"dos";"tres")

[] (corchetes)

- Permiten obtener un caracter de un texto
Ejemplo: a = "distrito-k"; a[3] devuelve "s"
- Permiten obtener un elemento de una lista (array)
Ejemplo: a = #("uno";"dos";"tres"); a[2] devuelve "dos"
- Evalúan una función predefinida (depende de la aplicación)
Ejemplo (sólo en Contasoft): [s '430'] devuelve el saldo de la cuenta 430

{ } (llaves)

Delimitan un bloque de código
Ejemplo: si a == b entonces {a = a + 1} sino {a = a - 1}

@ (arroba)

Permite obtener el valor de una variable cuyo nombre se obtiene a partir de una expresión
Ejemplo: a = 25; nombre = "a"; @nombre devuelve 25

& (ampersand)

Permite obtener la variable cuyo nombre se obtiene a partir de una expresión
Ejemplo: a = 25; &a devuelve la variable a, no el valor 25

;(punto y coma)

- Permite separar los elementos de una lista (array)
Ejemplo: a = #("uno";"dos";"tres")
- Permite separar los argumentos o parámetros de una función
Ejemplo: Min(a; b)
- Permite ejecutar más de una orden, separándolas entre sí. El valor devuelto por el conjunto de órdenes es el último evaluado.
Ejemplo: a = 3; b = 4; a + b; Se ejecutan las tres órdenes, y devuelve 7

6. Estructuras de lenguaje

funciones

Para llamar a una función, debe indicarse su nombre, y si desean indicarse argumentos o parámetros, deben indicarse a continuación entre paréntesis, separados entre sí por ";"

Ejemplo de llamada a una función sin argumentos: Pi

Ejemplo de llamada a una función con argumentos: Max(a;b)

si <expresión lógica> entonces <bloque de código>

Si <expresión lógica> devuelve cierto, ejecuta <bloque de código>

Ejemplo: si a < b entonces {"a es muy pequeño"}

si <expresión lógica> entonces <bloque de código 1> sino <bloque de código 2>

Si <expresión lógica> devuelve cierto, ejecuta <bloque de código 1>, pero si <expresión lógica> devuelve falso, ejecuta <bloque de código 2>

Ejemplo: si a < b entonces {"a es muy pequeño"} sino {"a está bien"}

mientras <bloque de código 1> haz <bloque de código 2>

Si <bloque de código 1> devuelve cierto, y mientras siga siéndolo, ejecuta <bloque de código 2>

Ejemplo: mientras {a < 5} haz {a = a + 1}

haz <bloque de código 1> mientras <bloque de código 2>

Ejecuta el <bloque de código 1>, y sigue ejecutándolo mientras el <bloque de Código 2> devuelva cierto

Ejemplo: haz {a = a + 1} mientras {a < 5}

Nota: La diferencia con la orden anterior es que <bloque de código> se ejecuta al menos una vez.

7. Constantes predefinidas

Constante	Tipo	Descripción
Nada	Nada	El valor nulo
Cierto	Lógico	El valor lógico cierto
Falso	Lógico	El valor lógico falso
TNada	Tipo	El tipo Nada
TNumero	Tipo	El tipo Número
TTexto	Tipo	El tipo Texto
TLogico	Tipo	El tipo Lógico
TBloque	Tipo	El tipo Bloque
TVariable	Tipo	El tipo Variable
TLista	Tipo	El tipo Lista
TTipo	Tipo	El tipo Tipo

8. Funciones matemáticas

Función	Resultado	Descripción
Pi	Número	El número Pi (3,1415...)
Abs(<n>)	Número	El valor absoluto (sin signo) de <n>
Redondeo(<n>)	Número	El entero más cercano a <n>
Defecto(<n>)	Número	El entero inmediatamente inferior a <n>
Exceso(<n>)	Número	El entero inmediatamente superior a <n>
Entero(<n>)	Número	La parte entera de <n>
Decimal(<n>)	Número	La parte decimal de <n>
Positivo(<n>)	Número	Si <n> es positivo, <n>; sino, 0
Negativo(<n>)	Número	Si <n> es negativo, <n>; sino, 0
RedondeoDec(<n1>;<n2>)	Número	<n1> redondeado a <n2> decimales
RedondeoFactor(<n1>;<n2>)	Número	<n1> redondeado a un múltiplo de <n2>
LogXY(<n1>;<n2>)	Número	Logaritmo de <n1> en base <n2>
Potencia(<n1>;<n2>)	Número	<n1> elevado a <n2>
Min(<n1>;<n2>)	Número	El menor valor de <n1> y <n2>
Max(<n1>;<n2>)	Número	El mayor valor de <n1> y <n2>

9. Funciones de manipulación de textos

Función	Resultado	Descripción
TxTamaño(<t>)	Número	Devuelve el número de caracteres de <t>
TxRellenazq(<t1>;<t2>;<n>)	Texto	Devuelve <t1> completado con el caracter <t2> por la izquierda hasta que tenga <n> caracteres en total
TxRellenaDer(<t1>;<t2>;<n>)	Texto	Devuelve <t1> completado con el caracter <t2> por la derecha hasta que tenga <n> caracteres en total
TxReemplaza(<t1>;<t2>;<t3>)	Texto	Sustituye en <t1> todas las apariciones del texto <t2> por el texto <t3>
TxSustituye(<t1>;<t2>;<t3>)	Texto	Sustituye en <t1> todas las apariciones de cada uno de los caracteres de <t2> por el caracter correspondiente de <t3> (el primero de <t2> por el primero de <t3>...)
TxLimpialzq(<t1>;<t2>)	Texto	Devuelve <t1> sin los caracteres <t2> que

		podiera tener por la izquierda
TxLimpiaDer(<t1>;<t2>)	Texto	Devuelve <t1> sin los caracteres <t2> que pudiera tener por la derecha
TxPosicion(<t1>;<t2>)	Número	Devuelve la posición que ocupa <t1> en <t2>. Si <t1> no aparece en <t2>, devuelve -1
TxTrozo(<t>;<n1>;<n2>)	Texto	Devuelve un trozo de <t> que comienza en <n1>, y tiene <n2> caracteres de longitud
TxBorra(<t>;<n1>;<n2>)	Texto	Devuelve <t>, luego de quitarle <n2> caracteres a partir de <n1>
TxCuantasPalabras(<t1>;<t2>)	Texto	Devuelve cuántas palabras contiene el texto <t1>, usando el primer carácter de <t2> como separador de palabras
TxPalabra(<t1>;<t2>;<n>)	Texto	Devuelve la palabra de posición <n>, empezando a contar en cero, del texto <t1>, usando el primer carácter de <t2> como separador de palabras
TxRestoPalabras(<t1>;<t2>;<n>)	Texto	Devuelve lo que haya en el texto <t1> a partir de la palabra de posición <n>, sin incluirla, empezando a contar en cero, usando el primer carácter de <t2> como separador de palabras
Intro	Texto	Devuelve un retorno de carro
Entrecomillar(<t1>;<t2>)	Texto	Devuelve un texto igual a <t1> añadiéndole por el principio y por el final la comilla que se especifique como <t2> y duplicándola cuando aparezca dentro del texto
Entrecomillar(<t>)	Texto	Entrecomilla utilizando la comilla simple

10. Funciones de conversión

Función	Resultado	Descripción
EnTexto(<x>)	Texto	Convierte <x> a texto
EnNumero(<t>)	Número	Convierte <t> a número, si es posible
EnLetraF(<n>)	Texto	<n1> expresado en letras, en femenino y sin decimales
EnLetraM(<n>)	Texto	<n1> expresado en letras, en masculino, con dos decimales, y finalizado en "céntimos" en caso de que haya decimales
EnLetra(<n>)	Texto	Es lo mismo que en EnLetraM(<n>)
Formatea(<n>)	Texto	Convierte <n> a texto

Formatea(<t>;<n>)	Texto	Convierte <n> a texto usando la máscara <t>
-------------------	-------	---

11. Funciones de manipulación de listas (arrays)

Función	Resultado	Descripción
LCrea	Lista	Crea y devuelve una lista vacía
LAñade(<lista>;<x>)	Lista	Añade <x> a la lista <lista>. Retorna la propia <lista>
LBorra(<lista>)	Nada	Elimina todos los items de <lista>
LTamaño(<lista>)	Número	Devuelve el número de items de <lista>
LQuita(<lista>;<n>)	Nada	Quita el item de índice <n> de <lista>
LAplica(<lista>;<bloque>)	Nada	Itera <lista>, asignando a la variable "params" el item actual, y ejecutando <bloque>
LCarga(<lista>;<t>)	Nada	Lee del fichero de nombre <t>, una lista, y la asigna a <lista>
LGraba(<lista>,<t>)	Nada	Escribe el contenido de <lista> en el fichero de nombre <t>
TipoLista(<lista>)	Lista	Devuelve una lista con los tipos de <lista>
ChequeaTipoltem(<lista>;<x>)	Nada	Verifica que todos los items de <lista> sean del tipo de <x>, o del tipo <x>

12. Funciones de manipulación de fechas

FechaHoy	Texto	Devuelve la fecha actual
HoraActual	Texto	Devuelve la hora en formato hh:nn:ss
HoraActual(<t>)	Texto	Devuelve la hora en formato <t>
FechaEnNúmero(<n>)	Número	Convierte una fecha en texto en una fecha numérica (permite sumar y restar fechas y sumarles y restarles días)
FechaEnTexto(<n>)	Texto	Convierte una fecha numérica a un fecha en texto (formato dd/mm/yyyy)
FormateaFecha(<t>)	Texto	Convierte la fecha a formato dd/mm/yyyy
FormateaFecha(<t1>;<t2>)	Texto	Convierte la fecha <t2> al formato especificado por <t1>
DiferenciaEnDías(<t1>;<t2>)	Número	Devuelve el número de días entre las fechas <t1> y <t2>

EsAñoBisiesto(<t>)	Lógico	Devuelve cierto o falso según el año de la fecha <t> sea bisiesto o no
EsMismoMes(<t1>;<t2>)	Lógico	Devuelve cierto o falso según las fechas <t1> y <t2> indiquen el mismo mes del mismo año
DíasMes(<t>)	Número	Devuelve el número de días del mes indicado por la fecha <t>
DíasAño(<t>)	Número	Devuelve el número de días del año indicado por la fecha <t>
AnteriorMes(<t>)	Texto	Devuelve la fecha correspondiente al último día del mes anterior a la fecha <t>
SiguienteMes(<t>)	Texto	Devuelve la fecha correspondiente al primer día del mes siguiente a la fecha <t>
PrimerDíaMes(<t>)	Texto	Devuelve la fecha correspondiente al primer día del mes indicado por la fecha <t>
ÚltimoDíaMes(<t>)	Texto	Devuelve la fecha correspondiente al último día del mes indicado por la fecha <t>
DíaDe(<t>)	Número	Devuelve el día del mes (de 1 a 31) de la fecha <t>
MesDe(<t>)	Número	Devuelve el mes (de 1 a 12) de la fecha <t>
AñoDe(<t>)	Número	Devuelve el año de la fecha <t>
SemanaDe(<t>)	Número	Devuelve la semana dentro del año (de 1 a 54) en que se encuentra situada la fecha <t>
TrimestreDe(<t>)	Número	Devuelve el trimestre (de 1 a 4) en que se encuentra situada la fecha <t>

13. Funciones de manipulación de variables

Función	Resultado	Descripción
VerVars(<t1>;<t2>)	Nada	Muestra un cuadro de diálogo con el título <t1>, que permite ver el valor de la lista de variables indicadas en <t2>, separadas por ";", y aceptando el uso del comodín "*". Si no se especifica <t2>, todas las variables.
EditarVars(<t1>;<t2>)	Nada	Muestra un cuadro de diálogo con el título <t1>, que permite ver y modificar el valor de la lista de variables indicadas en <t2>, separadas por ";", y aceptando el uso del comodín "*". Si no se especifica <t2>, todas las variables.
DuplicarVars(<t1>;<t2>)	Nada	Duplica, agregándoles el prefijo <t2>, la lista de variables indicadas en <t1>, separadas por ";", y aceptando el uso del comodín "*"

BorrarVars(<t>)	Nada	Borra la lista de variables indicadas en <t>, separadas por ";", y aceptando el uso del comodín "*"
RenombrarVars(<t1>;<t2>)	Nada	Renombra las variables indicadas en <t1>, asignándoles los nombres indicados en <t2>
SustituirVars(<t1>;<t2>)	Nada	Sustituye los valores de las variables indicadas
SustituirVars(<t1>;<t2>;<t3>)	Nada	Sustituye los valores de las variables indicadas, y las renombra
SetVariablePrefix(<t>)	Nada	Fija <t> como prefijo por defecto para las nuevas variables
GetVariablePrefix	Texto	Devuelve el prefijo por defecto para las nuevas variables
Tipo(<x>)	Tipo	Devuelve el tipo de <x>

14. Funciones de acceso a base de datos

Función	Resultado	Descripción
ObtListaSQL(<t>)	Lista	Ejecuta la sentencia SQL indicada por <t>, que debe ser un select que recupere una única columna, y devuelve una lista que contiene un elemento de tipo texto con la columna de cada uno de los registros recuperados
CuantosRegistros(<t1>;<t2>)	Número	Devuelve qué cantidad de registros de la tabla <t1> cumplen con la condición <t2>, que debe tener la misma sintaxis que un where de SQL

15. Funciones varias

Función	Resultado	Descripción
Mensaje(<t>)	Nada	Muestra un cuadro de diálogo informativo, con el mensaje <t>, y un botón "Aceptar"
Pregunta(<t1>;<t2>)	Texto	Muestra un cuadro de diálogo solicitando un texto, con el mensaje <t1>, y los botones "Aceptar" y "Cancelar". Si se especifica <t2>, éste se mostrará como texto por defecto. Si se selecciona "Aceptar", la función devuelve el texto introducido. Si se selecciona "Cancelar", interrumpe el proceso
Confirma(<t>)	Lógico	Muestra un cuadro de diálogo de confirmación, con el mensaje <t>, y los botones "Sí", "No" y "Cancelar". Si se selecciona "Sí", la función devuelve Cierto. Si se selecciona "No", la función devuelve Falso. Si se selecciona

		"Cancelar", interrumpe el proceso
Error(<t>)	Nada	Muestra un cuadro de diálogo de error, con el mensaje <t>, y el botón "Aceptar". Interrumpe el proceso
SiguienteControl(<n>)	Nada	Avanza el foco de teclado (el control que recibe las pulsaciones del teclado) al control que esté <n> veces después que el actual
AnteriorControl(<n>)	Nada	Avanza el foco de teclado (el control que recibe las pulsaciones del teclado) al control que esté <n> veces antes que el actual
Imprimir(<t>)	Nada	Imprime el formato indicado por <t> desde el formato actual
PararImpresion	Nada	Para de imprimir el formato actual
ResultadoNada()	Nada	Evalúa el bloque de código (especificado entre llaves) y descarta el valor de resultado, devolviendo "nada"

Formato de máscaras para números

Una máscara para números es un texto que indica la forma de presentar un número, y puede estar formada por los siguientes especificadores:

- "0"** Indica la posición de un dígito. Si el valor que se está formateando tiene un dígito en la posición donde el "0" aparece en la máscara, entonces se muestra ese dígito. Si no, se muestra un "0" en esa posición.
- "#"** Indica la posición de un dígito. Si el valor que se está formateando tiene un dígito en la posición donde el "0" aparece en la máscara, entonces se muestra ese dígito. Si no, no se muestra nada en esa posición.
- ","** Separador decimal. El primer carácter "." en la máscara determina la posición del separador decimal en el valor presentado. Cualquier "." adicional se ignora.
- ","** Separador de miles. Si la máscara contiene algún carácter ",", el valor presentado tendrá separadores de miles cada tres dígitos a la izquierda del separador decimal. La ubicación y la cantidad de "," en la máscara no afecta al resultado.
- "e+"** Notación científica. Si cualquiera de los textos "e+", "e-" aparecen en la máscara, el número es presentado utilizando notación científica. Puede agregarse a estos símbolos un grupo de hasta cuatro "0", para indicar el mínimo número de dígitos del exponente. El especificador "e+" muestra siempre el signo del exponente, ya sea positivo o negativo, pero el especificador "e-" sólo muestra el signo si es negativo.
- 'xx' ó "xx"** Los caracteres encerrados por comillas simples o dobles se muestran tal como se indican, y no afectan al formato. Lo mismo ocurre si utilizamos cualquier carácter que no sea un especificador de la presente lista.
- ";"** Separa las secciones para números positivos, negativos o ceros en la máscara.

Notas

- El número se presenta siempre redondeado a la máxima cantidad de decimales indicados por los "0" ó "#" a la derecha del separador decimal. Si la máscara no contiene el separador decimal, el número se redondeará al entero más cercano.
- Si el número tuviera más dígitos que los indicados en la máscara, estos dígitos se presentarán a la izquierda del primer "0" ó "#".
- Para permitir diferentes máscaras para valores positivos, negativos y ceros, la máscara puede contener entre una y tres secciones separadas por ";":
 - Una sección:
La máscara se aplica a todos los valores
 - Dos secciones:
La primera se aplica a valores positivos y ceros, y la segunda a negativos
 - Tres secciones:
La primera se aplica a valores positivos, la segunda a negativos, y la tercera a ceros
- Si la máscara está vacía, o no se especifica máscara, el valor se presenta utilizando la siguiente máscara por defecto: ",0.#####". Es decir, con separador de miles, y con hasta cinco decimales, presentando siempre el cero para valores menores a uno.
- En cualquier caso, si hubiera más de 18 dígitos a la izquierda del separador decimal, se presentará el número en notación científica, aunque no se haya especificado.

Ejemplos

Valores	1234	-1234	0,5	0
""	1234	-1234	0,5	0
"0"	1234	-1234	1	0
"0.00"	1234,00	-1234,00	0,50	0,00
"#.##"	1234	-1234	,5	
"#,##0.00"	1.234,00	-1.234,00	0,50	0,00
"#,##0.00;(#,##0.00)"	1.234,00	(1.234,00)	0,50	0,00
"#,##0.00;;Cero"	1.234,00	-1.234,00	0,50	Cero
"0.000E+00"	1,234E+03	-1,234E+03	5,000E-01	0,000E+00
"#.###E-0"	1,234E3	-1,234E3	5E-1	0E0

Formato de máscaras para fechas

Una máscara para fechas es un texto que indica la forma de presentar una fecha, y puede estar formada por los siguientes especificadores:

d	Muestra el día como un número de uno o dos dígitos (1-31)
dd	Muestra el día como un número de dos dígitos (01-31)
ddd	Muestra el nombre abreviado del día de la semana (lun-dom)
dddd	Muestra el nombre completo del día de la semana (lunes-domingo)
dddddd	Muestra la fecha con formato dd/mm/yyyy
dddddd	Muestra la fecha usando el formato de fecha larga definido en Windows Normalmente: dddd, dd' de 'mmmm' de 'yyyy
m	Muestra el mes como un número de uno o dos dígitos (1-12)
mm	Muestra el mes como un número de dos dígitos (01-12)
mmm	Muestra el nombre abreviado del mes (ene-dic)
mmmm	Muestra el nombre completo del mes (enero-diciembre)
yy	Muestra el año como un número de dos dígitos (00-99)
yyyy	Muestra el año como un número de cuatro dígitos (0000-9999)
/	Muestra el separador de fecha que se haya definido en Windows (normalmente la barra, "/")
'xx' ó "xx"	Los caracteres encerrados por comillas simples o dobles se muestran tal como se indican, y no afectan al formato.

Formato de máscaras para horas

Una máscara para horas es un texto que indica la forma de presentar una hora, y puede estar formada por los siguientes especificadores:

h	Muestra la hora del día como un número de uno o dos dígitos (0-23)
hh	Muestra la hora del día como un número de dos dígitos (00-23)
n	Muestra el minuto como un número de uno o dos dígitos (0-59)
nn	Muestra el minuto como un número de dos dígitos (00-59)
s	Muestra el segundo como un número de uno o dos dígitos (0-59)
ss	Muestra el segundo como un número de dos dígitos (00-59)
t	Muestra la hora con formato "hh:nn"
tt	Muestra la hora con el formato definido en Windows (normalmente "h:nn:ss")
am/pm	Usa formato de 12 horas con el especificador h o hh precedente, y muestra 'am' o 'pm' según corresponda. El am ó pm del resultado se mostrará en mayúsculas, minúsculas o ambas según estén en el especificador.
a/p	Usa formato de 12 horas con el especificador h o hh precedente, y muestra 'a' o 'p' según corresponda. El a ó p del resultado se mostrará en mayúsculas o minúsculas según esté el especificador.
:	Muestra el separador de hora que se haya definido en Windows (normalmente los dos puntos, ":")
'xx' ó "xx"	Los caracteres encerrados por comillas simples o dobles se muestran tal como se indican, y no afectan al formato.

Funciones específicas de Contasoft

Función	Resultado	Descripción
PreguntaMeses	Texto	Muestra un cuadro de diálogo solicitando los períodos, y devuelve una lista de períodos, que puede utilizarse en la función [] (ver más abajo)
DatoEmpresa(<t>)	Varía	Devuelve el valor del campo <t> de la tabla de configuración de la empresa
Formula(<n>)	Varía	Localiza la fórmula con el código <n> en la tabla de fórmulas, y la evalúa
Formula(<t>)	Varía	Localiza la fórmula con el nombre <t> en la tabla de fórmulas, y la evalúa

[<c>]	Número	Saldo de la lista de cuentas <c>
[D H S <c> <p>]	Número	Debe, haber o saldo de la lista de cuentas <c> en la lista de períodos <p>. Si no se especifica <p>, se asume el total del ejercicio
[N <c>]	Texto	Nombre de la cuenta <c>
[C <c>]	Número	Valor de la lista de casillas <c> (en impresos)

Listas de cuentas

Los elementos que conforman una lista de cuentas, son los siguientes:

<n>	Cuenta	Ejemplo: "400"
<n>-<n>	Rango de cuentas	Ejemplo: "400-430"

Puede anteponerse a una cuenta o rango uno de los siguientes modificadores:

-	Devuelve el resultado, cambiado de signo
P	Devuelve el resultado, sólo si es positivo
N	Devuelve el resultado, sólo si es negativo

La lista de cuentas estará formada por una o más cuentas o rangos, separados por ";".

Ejemplos

"430"	Cuenta 430
"400-430"	Cuentas de la 400 a la 430
"400-430;-472"	Cuentas de la 400 a la 430, más la cuenta 472 cambiada de signo
"P400-430;472"	Cuentas de la 400 a la 430 sólo si son positivas, más la cuenta 472

Listas de períodos

Los elementos que conforman una lista de períodos, son los siguientes:

ANT	: Resultado del ejercicio anterior
AP	: Asiento de apertura
1 a 12	: Meses del año, 1=Enero, 2=Febrero, ..., 12=Diciembre
REX	: Asiento de regularización de existencias
REG	: Asiento de regularización
CIE	: Asiento de cierre

La lista de períodos puede estar formada por uno o más elementos, separados por ";", o por un rango de elementos, separados por "-".

Ejemplos

"AP;1;3"	: Asiento de apertura, enero, marzo
"AP-REG"	: Desde asiento de apertura hasta regularización
""	: Todos los períodos
","	: Ningún período

Funciones específicas de Pymesoft para impresión de documentos

El sistema de impresión de documentos de Pymesoft proporciona dos objetos que permiten indicar fórmulas, definiendo todas las variables indicadas en la parte derecha del editor de formularios. Estos objetos son:

DocumentoFórmula

La fórmula se evaluará una vez para cada objeto y página. Si existen varias órdenes, separadas por ";", se procederá normalmente, devolviendo el resultado de la última orden. Las funciones adicionales disponibles en este contexto son:

DatoAdicional Se explica más adelante

ArtículoFórmula

La fórmula se evaluará una vez para cada línea del documento. Si existen varias órdenes, separadas por ";", se evaluará cada orden por separado, agregando el resultado de cada una al cuerpo del documento. Las funciones adicionales disponibles en este contexto son:

DatoAdicional Se explica más adelante

ArtículoCaracterística Se explica más adelante

ArtículoDocOrigenDatoAdicional Funciona como DatoAdicional, pero obtiene los datos del documento de origen de la línea actual

CambióOrigen Si el origen de la línea actual ha cambiado con respecto a la anterior, la función devuelve Cierto, y si no, devuelve Falso

F<n> Cambia la fuente actual a la de índice <n>

I<n> Cambia la indentación actual a <n> centímetros

S<n> Inserta <n> líneas a esta columna, e inserta líneas en blanco a todas las demás columnas hasta quedar a la altura de ésta.

#ENTER Inserta una línea en blanco

#ORIGEN Si el origen de esta línea ha cambiado con respecto a la anterior, devuelve: "De <tipo > N° <serie >-<número> del <fecha >:"

DatoAdicional(<entidad>;<formato>;<dato>)

Esta función permite obtener información sobre los datos adicionales de las entidades. Los parámetros son:

<entidad>: Un texto que indica la entidad de la se desea obtener información

Documento	Dato adicional del documento actual
Línea	Dato adicional de la línea actual del documento
Artículo	Dato adicional del artículo de la línea actual del documento
Cliente	Dato adicional del cliente o proveedor del documento actual
Obra	Dato adicional de la obra actual
ArtículoObra	Dato adicional del artículo de la línea actual de la obra

<formato>: Un texto que indica el formato del resultado. Las macros válidas son:

%n	El nombre del dato adicional
%v	El valor del dato adicional

<dato>: Un texto opcional que indica el nombre del dato adicional requerido. Si este parámetro no se indica, la función devuelve una lista con todos los datos existentes.

Ejemplos

DatoAdicional("Cliente"; "El valor de %n es %v") podría devolver la siguiente lista

- El valor de Ultima visita es 12/01/2001
- El valor de Garantía es Sí
- El valor de Importe de la póliza es 12.500

DatoAdicional("Artículo"; "%v"; "Garantía") podría devolver el siguiente valor

6 meses

Funciones equivalentes

Para una mayor comodidad de uso, se han definido las siguientes funciones equivalentes:

En documentos

DocumentoDatoAdicionalNombre(<dato>)	= DatoAdicional("Documento"; "%n"; <dato>)
DocumentoDatoAdicionalValor(<dato>)	= DatoAdicional("Documento"; "%v"; <dato>)
ClienteDatoAdicionalNombre(<dato>)	= DatoAdicional("Cliente"; "%n"; <dato>)
ClienteDatoAdicionalValor(<dato>)	= DatoAdicional("Cliente"; "%v"; <dato>)
ArtículoDatoAdicionalNombre(<dato>)	= DatoAdicional("Línea"; "%n"; <dato>)
ArtículoDatoAdicionalValor(<dato>)	= DatoAdicional("Línea"; "%v"; <dato>)

En órdenes de fabricación

DatoAdicionalNombre(<dato>)	= DatoAdicional("Obra"; "%n"; <dato>)
DatoAdicionalValor(<dato>)	= DatoAdicional("Obra"; "%v"; <dato>)
ArtículoDatoAdicionalNombre(<dato>)	= DatoAdicional("ArtículoObra"; "%n"; <dato>)
ArtículoDatoAdicionalValor(<dato>)	= DatoAdicional("ArtículoObra"; "%v"; <dato>)

ArtículoCaracterística(<formato>; <dato>; <dimensión1>; ...; <dimensiónN>; <orden>)

Esta función permite obtener información sobre las características indicadas en las líneas de los documentos. Los parámetros son:

<formato>: Un texto que indica el formato del resultado. Las macros válidas son:

- %n**: El nombre de la característica. Ejemplo: "Lote" ó "Talla-Color"
- %v**: El valor de la característica. Ejemplo: "20" ó "38-Negro"
- %p**: El par nombre-valor. Ejemplo: "Lote 20" ó "Talla 38, Color Negro"
- %c**: La cantidad indicada. Ejemplo: "5,2"
- %u**: La unidad en la que se expresa dicha cantidad. Ejemplo: "Kg"

<dato>: Un texto opcional que indica el nombre de la característica requerida. Si este parámetro no se indica, la función devuelve una lista con todas las características existentes en la línea del documento.

<dimensión1> ... <dimensiónN>: Textos opcionales que indican el valor requerido de cada una de las dimensiones de la característica. Si se omite alguna dimensión (o todas), la función devolverá todos los valores existentes en esa dimensión.

<orden>: Texto opcional que indica el nombre de la dimensión que devolverá y por la que se ordenará la lista resultante.

Ejemplos

ArtículoCaracterística("%p") podría devolver la siguiente lista

- Talla 36, Color Rojo
- Talla 36, Color Negro
- Talla 38, Color Negro
- Lote 18
- Lote 20
- Lote 22

ArtículoCaracterística("%p : %c", "Talla-Color") podría devolver la siguiente lista

- Talla 36, Color Rojo : 12,5
- Talla 36, Color Negro : 27,0
- Talla 38, Color Negro : 11,4

ArtículoCaracterística("%n %v = %c %u", "Lotes") podría devolver la siguiente lista

- Lote 18 = 5,2 Kg
- Lote 20 = 12,4 Kg
- Lote 22 = 0,7 Kg

ArtículoCaracterística("%p", "Talla-Color", "36") podría devolver la siguiente lista

- Talla 36, Color Rojo
- Talla 36, Color Negro

ArtículoCaracterística("%p", "Talla-Color", "", "Negro") podría devolver la siguiente lista

- Talla 36, Color Negro
- Talla 38, Color Negro

ArtículoCaracterística("%p", "Talla-Color", "", "", "Color") podría devolver la siguiente lista

- Talla 36, Color Negro
- Talla 38, Color Negro
- Talla 36, Color Rojo

Funciones equivalentes

Para una mayor comodidad de uso, se han definido las siguientes funciones equivalentes:

ArtículoCaracterísticaNombre(...)	= ArtículoCaracterística("%n"; ...)
ArtículoCaracterísticaValor(...)	= ArtículoCaracterística("%v"; ...)
ArtículoCaracterísticaPar(...)	= ArtículoCaracterística("%p"; ...)
ArtículoCaracterísticaCantidad(...)	= ArtículoCaracterística("%c"; ...)
ArtículoCaracterísticaUnidad(...)	= ArtículoCaracterística("%u"; ...)