

Elemento Regla	Ejemplo
Aplicar Cuando	<code>NOT goIntegracion.plInventarioPermanente OR (NOT goIntegracion.plInventarioPermanente AND NOT goIntegracion.plInventarioDetallado)</code>
	<code>!linv_perm or (linv_perm and !linv_deta)</code>
Cuenta Contable	<code>mObtenerCuentaContable(Contable_Facturas[7]; Contable_Clientes[1]; Cuenta_Integracion["11301"]; "4101001")</code>
	<code>iif(!empty(Inf_doc("vfactura","c",7)),Inf_doc("vfactura","c",7),iif(empty(Inf_tab("clientes","c",1)),Inf_tab("clientes","c",1),iif(empty(cta_int(11301)),cta_int(11301),4101001)))</code>
Cuenta de Gastos	<code>mObtenerCuentaGasto(Contable_Facturas[1]; Cuenta_Integracion["11301"]; loCuentaContable.pcCuentaGastos, "otr_gas")</code>
	<code>iif(!empty(Inf_doc("vfactura","g",1)),Inf_doc("vfactura","g",1),sccuenta.co_gas)</code>
Centros de Costo	<code>mObtenerCentrosCosto(Contable_Facturas[1]; Cuenta_Integracion["11301"]; loCuentaContable.pcCentrosCostos, "otr_cos")</code>
	<code>iif(!empty(Inf_doc("vfactura","d",1)),Inf_doc("vfactura","d",1),"["+sccuenta.centro_co+";100]")</code>
Monto del Asiento	<code>SII({{fiscal}}; {{mon_net}}; {{mon_bru}})</code>
	<code>iif(vfactura.contrib,vfactura.tot_bruto,vfactura.tot_bruto-Calcula_IVA("vfactura"))</code>
Descripción del Asiento	<code>"VENT." & RecortarTexto({{cod_cli}}) & "/" & TextoIzquierda({{nom_cli}};34) & " Regla N°: " & RecortarTexto(goRegla.Codigo)</code>
	<code>"VENT."+alltrim(vfactura.co_cli)+"/"+left(clientes.cli_des,34)+" Regla N°: "+alltrim(scint_ti.co_int_ti)</code>
Otros Campos	
Fecha	<code>Ahora ()</code>
	<code>SumarComponenteFecha ({{fec_ini}};-30; "DIA")</code>
	<code>NuevaFecha (ObtenerAño ({{fec_fin}}); ObtenerMes ({{fec_fin}}); 1)</code>
Nivel del Documento	<code>{{nivel}}</code>
	<code>goSucursal.pnNivel</code>
	<code>Minimo (goUsuario.pnNivel+10;100)</code>
Moneda	<code>{{cod_mon}}</code>
	<code>goEmpresa.pcMonedaBase</code>
	<code>goEmpresa.pcMonedaAdicional</code>
Tasa Cambiaria	<code>{{tasa}}</code>
Auxiliar	<code>mObtenerAuxiliar ({{cod_aux}}; Cuenta_Integracion["11301"]; loCuentaContable.pcAuxiliar; "otr_aux")</code>
Clasificador	<code>SII ({{nivel}}=>80; "Importante"; NoVacio ({{cod_cla}}; loCuentaContable.pcClasificador)</code>

Nomenclatura de las Referencias:

En este manual de referencia los elementos en color **gris** representan indicadores de formato, pero no son parte de la sintaxis de los comandos (y no deben ser incluidos en las reglas de integración).

Los Corchetes (`[` y `]`) representan elementos opcionales. Una barra vertical representa dos valores posibles mutuamente excluyentes. Un asterisco (*) a la derecha de un elemento o grupo indica que el elemento o grupo es opcional y puede aparecer cualquier cantidad de veces, un signo de suma (+) actúa como el asterisco, pero indica que el elemento o grupo debe aparecer al menos una vez. Las Llaves (`{M,N}`) a la derecha de un elemento o grupo indican que éste debe repetirse al menos M y no más de N veces.

Ejemplo	Descripción
<code>A B</code>	Se requiere A o B .
<code>[A B]</code>	Se permite A o B , pero son opcionales.
<code>A[,B]</code>	Se requiere un A y se permite B pero es opcional. E.g. " A,B " y " A " son válidos.
<code>A,B{0,2}</code>	Se requiere un A y hasta dos B . E.g. " A ", " A,B " y " A,B,B " son válidos, pero " A,B,B,B " no lo es.
<code>[A,]*B</code>	A es opcional, pero puede aparecer una o más veces (seguido de coma). B es obligatoria al final.
<code>[A[, (B,){2,3}]]C</code>	A es opcional, pero si se especifica puede estar seguido de una coma y 2 o 3 B (cada uno seguido de coma), pero debe terminar en C . e.g. " C ", " A,C " y " A,B,B,C " son válidos, pero " A,B,B " y " A,B,C " no.

Objetos Globales y Locales Disponibles

<p>goEmpresa: Hace referencia a la empresa de la Contabilidad desde la cual se hace la integración.</p>	<ul style="list-style-type: none"> • <code>goEmpresa.pcCodigo</code> • <code>goEmpresa.pcNombre</code> • <code>goEmpresa.pcMonedaBase</code> • <code>goEmpresa.pcMonedaAdicional</code> • <code>goEmpresa.plUsaInventarioPermanente</code>: Indica si la empresa maneja inventario permanente. • <code>goEmpresa.plUsaInventarioDetallado</code>: Indica si la empresa maneja inventario permanente detallado.
<p>goSucursal: Hace referencia a la sucursal de la Contabilidad desde la cual se hace la integración.</p>	<ul style="list-style-type: none"> • <code>goSucursal.pcCodigo</code> • <code>goSucursal.pcNombre</code> • <code>goSucursal.pnNivel</code>
<p>goUsuario: Hace referencia al usuario de la Contabilidad que está ejecutando la integración.</p>	<ul style="list-style-type: none"> • <code>goUsuario.pcCodigo</code> • <code>goUsuario.pcNombre</code> • <code>goUsuario.pnNivel</code> • <code>goUsuario.pcCorreo</code>
<p>loIntegracion: Hace referencia a algunos parámetros del documento de integración que inició la integración del administrativo con la contabilidad.</p>	<ul style="list-style-type: none"> • <code>loIntegracion.pdFecha</code>: Fecha del documento. • <code>loIntegracion.pdFechaDesde</code>: Inicio de la integración. • <code>loIntegracion.pdFechaHasta</code>: Fin de la integración. • <code>loIntegracion.pcTipoAgrupacion</code>: Indica el tipo de agrupación de la integración actual. • <code>loIntegracion.pcTipoCosto</code>: Indica el criterio de costeo ("Ultimo" o "Promedio") de la integración actual. • <code>loIntegracion.pnNivel</code>: nivel del documento de integración.
<p>loRegla: Hace referencia a algunas propiedades de la regla de integración actual.</p>	<ul style="list-style-type: none"> • <code>loRegla.pcCodigo</code>: Código de la regla. • <code>loRegla.pcNombre</code>: Nombre de la regla. • <code>loRegla.pcTipoMovimiento</code>: Tipo de movimiento de la regla ("Debe" o "Haber"). • <code>loRegla.pnTipoDocumento</code>: Tipo de documento asociado a la regla. • <code>loRegla.pnNivel</code>: Nivel de la regla.
<p>loCuentaContable: Hace referencia a algunas propiedades de la <i>Cuenta Contable Previamente Seleccionada</i>. La Cuenta Contable queda determinada justo después de que se establece si la regla actual se ejecutará o no (según el campo <i>Ejecutar Cuando</i>).</p>	<ul style="list-style-type: none"> • <code>loCuentaContable.pcCodigo</code>: Código de la <i>Cuenta Contable</i>. • <code>loCuentaContable.pcNombre</code>: Nombre de la <i>Cuenta Contable</i>. • <code>loCuentaContable.plManejaCuentaGastos</code>: Indica si la <i>Cuenta Contable</i> actual maneja <i>Cuenta de Gastos</i>. • <code>loCuentaContable.pcCuentaGastos</code>: Código de la <i>Cuenta</i>

	<p>de Gastos asociada a la <i>Cuenta Contable</i> actual.</p> <ul style="list-style-type: none"> • loCuentaContable.plManejaAuxiliar: Indica si la <i>Cuenta Contable</i> actual maneja <i>Auxiliar</i>. • loCuentaContable.pcAuxiliar: Código del <i>Auxiliar</i> asociado a la <i>Cuenta Contable</i> actual. • loCuentaContable.pcCentrosCostos: Devuelve el código del centro de costos predeterminado asociado a la <i>Cuenta Contable</i> actual. • loCuentaContable.plMovimientos: Indica si la <i>Cuenta Contable</i> es de movimientos (detalles) o no.
--	--

Valores Literales:

Tipo	Formato	Ejemplos		
Cadena	"[^"]*" '[^']*'	""	"Cadena Válida ñ"	'_(válida'€) 3+6'
Numero	d+[.d*] .[d]+	1234	12.345	.123
Fecha	#AAAAMMDD[HH:mm[:ss[.fff]]#	#20083112#		#20083112 01:05#
		#20083112 23:59#		#20083112 23:59:59#
		#20083112 23:59:59.999#		
Lógico	TRUE FALSE	TRUE	FALSE	
Nulo	NULL	NULL		
No Número	NAN	NAN		

Observaciones:

- Los literales de cadena pueden estar delimitados por comillas simples o dobles.
- Un literal de cadena delimitado por comillas simples no puede contener comillas simples.
- Un literal de cadena delimitado por comillas dobles no puede contener comillas dobles.
- Ningún literal de cadena puede contener caracteres de "Salto de Línea" (ASCII 10) ni "Retorno de carro" (ASCII 13).
- Para usar comillas simples, comillas dobles, avance de línea, retorno de carro, y por comodidad tabulaciones, en un literal de cadena se admiten las siguientes secuencias de escape:

Caracter	Código de Escape	Ejemplo	Salida
'	\q	'La mejor opción: \qeFactory\q'	La Mejor Opción: `eFactory`
"	\Q	"La mejor opción: \QeFactory\Q"	La Mejor Opción: "eFactory"
ENTER	\n \n\r	"La mejor opción:\n\r eFactory"	La Mejor Opción: eFactory
TAB	\t	"La mejor opción:\teFactory"	La Mejor Opción: eFactory

- **Nota:** En sistemas Linux, Unix, Amiga, MAC y similares se utiliza el carácter de Salto de Línea (ASCII 10) como estandar para "Nueva Línea" o "ENTER"; en DOS y Windows, por extensión de las viejas máquinas de escribir que dieron origen al teclado, se utiliza una combinación de dos caracteres para indicar una "Nueva Línea" o "Enter": un Salto de Línea (ASCII 10) seguido de un Retorno

de Carro (ASCII 13). Se recomienda usar solo la primera forma, es decir, para insertar un "Enter" en un literal de cadena se recomienda hacerlo como sigue: "La mejor opción:\neFactory"

Operadores

Aritméticos: solo esperan operandos numéricos o tratan todos los operandos como números. El resultado esperado es Numérico, pero algunas operaciones devuelven **NAN** o **NULL**.

Símbolo	Operacion	Ejemplo
^	Potencia	$2^5 = 32$
+	Identidad (no hace nada)	$+4.2 = 4.2$
	Suma	$5 + 4.2 = 9.2$
-	Negativo	$-5 = -5$
	Resta	$5 - 4.2 = 0.8$
*	Multiplicación	$5 * 4.2 = 21.0$
/	División	$7.1 / 4 = 1.75$
DIV	División Entera $A \text{ DIV } B = \text{Piso}(A / B)$	$7.1 \text{ DIV } 4 = 1$
MOD	División Modular (Resto) $A \text{ MOD } B = A - B * \text{Piso}(A / B)$	$7.1 \text{ MOD } 4 = 3.1$
&	Concatenación	<code>"alfa " & "beta" = "alfa beta"</code> <code>2.5 & ":" & NULL & TRUE = "2.5:true"</code>

Observaciones:

- Las operaciones numéricas no válidas (ej. $0/0$) devuelven **NAN**.
- Si uno de los operandos es **NAN**, o es de tipo Fecha o Cadena o el resultado es **NAN**.
- El valor especial **NULL** es evaluado como 0.
- El valor lógico **TRUE** es evaluado como 1. El valor lógico **FALSE** es evaluado como 0.
- El operador de concatenación acepta cualquier tipo de operando, pero si no son cadenas los convierte de acuerdo a las siguientes reglas:
 - Los valores de tipo Decimal se representan con un punto como separador decimal y sin separador de miles, e.g. `"1234.56"`.
 - Los valores de tipo fecha se representan con el formato completo, e.g. `"YYYYMMDD HHmmss.fff"`.
 - Los valores Lógicos **TRUE** y **FALSE** se representan por su nombre en minúsculas: `"true"` `"false"`
 - Los valores **NAN** y **NULL** se representan por una cadena vacía: `""`.
- Aunque el operador de concatenación haga una conversión automática es recomendable usar una de las funciones de formato.

De Comparación: Pueden manejar varios tipos de operandos. El resultado esperado es Lógico, pero algunas comparaciones devuelven **NAN** o **NULL**.

Símbolo	Comparación	Ejemplo
==	Igualdad	3.0 == 3 → TRUE
!=	Desigualdad	3.0 != 3 → FALSE
>	Mayor que	#20090101# > #20080101# → TRUE
<	Menor que	"alfa" < "beta" → TRUE
>=	Mayor o igual que	4 >= 4 → TRUE
<=	Menor o igual que	8.5 <= 4 → FALSE
LIKE	Parecido a	"alfabeto" LIKE "a[a-z]{2}b" → TRUE

Observaciones:

- Si los operandos son ambos de tipo Decimal, Cadena o Fecha son comparados de forma habitual.
- En las cadenas, el orden depende del código ASCII de los caracteres, eg. "A" < "a" y "." < "5".
- Si los operandos son lógicos, se considera **TRUE > FALSE**.
- Para los valores especiales **NAN** y **NULL**, se considera **NAN==NAN**, **NULL==NULL** y **NULL<NAN**.
- Al comparar valores de tipo Decimal con valores tipo Lógico, el valor lógico se considera 1 si es **TRUE** y 0 si es **FALSE**.
- El operador **LIKE** considera **NAN=="** y **NULL=="**. Si alguno de los operandos es de tipo Decimal, Fecha o Lógico, siempre devuelve **FALSE**. El segundo operando debe ser una *Expresión Regular* válida.
- Los casos de comparación (excepto para **LIKE**) se resumen en la siguiente tabla:

Tipo	Decimal	Cadena	Lógico	Fecha	NAN	NULL
Decimal	OK	devuelve NAN	devuelve NAN	devuelve NAN	devuelve NAN	NULL == 0
Cadena	devuelve NAN	OK	LEN(C) > 0 → TRUE LEN(C) == 0 → FALSE	devuelve NULL	NAN == ""	NULL == ""
Lógico	devuelve NAN	LEN(C) > 0 → TRUE LEN(C) == 0 → FALSE	TRUE > FALSE	devuelve NULL	NAN == FALSE	NULL == FALSE
Fecha	devuelve NAN	devuelve NULL	devuelve NULL	OK	devuelve NAN	NULL == #19000101#
NAN	devuelve NAN	NAN == ""	NAN == FALSE	devuelve NAN	NAN == NAN	NAN > NULL
NULL	NULL == 0	NULL == ""	NULL == FALSE	NULL == #19000101#	NAN > NULL	NULL == NULL

Lógicos: solo esperan operandos de tipo lógico o tratan todos los operandos como lógicos. El resultado de estas operaciones es siempre de tipo lógico.

Símbolo	Operacion	Ejemplo
NOT	Negación Lógica	NOT (5 > 3) → FALSE
AND	Conjunción Lógica	(5 >= 3) AND (3 <= 3) → TRUE
XOR	Disyunción Lógica Exclusiva	TRUE XOR (5 > 3) → FALSE
OR	Disyunción Lógica	TRUE OR (5 > 3) → TRUE

Observaciones:

- Al usar operadores lógicos, los valores de tipo no lógico son convertidos automáticamente según la siguiente tabla:

Tipo	Valor Lógico
Decimal	Sii (D <> 0; TRUE; FALSE)
Cadena	Sii (Longitud(C) > 0; TRUE; FALSE)
Fecha	Sii (F > #19000101#; TRUE; FALSE)
NAN	FALSE
NULL	FALSE

Funciones Especiales

Función: mObtenerCuentaContable (Param1 [, ParamN] *) → CADENA

Contexto	Solo es válida en el campo "Cuenta Contable" de la regla de integración.
Parámetros	<p>De uno a diez de los siguientes:</p> <ul style="list-style-type: none"> Contable_NombreTabla [N]: Referencia a la <i>Cuenta Contable</i> almacenada en la página N° "N" en el campo de información contable {{Contable_NombreTabla}}. Cuenta_Integracion["CODIGO"]: Referencia a la <i>Cuenta Contable</i> relacionada a la <i>Cuenta de Integración</i> de código "CODIGO". {{Nombre_Campo}}: Indica que el código de la <i>Cuenta Contable</i> se leerá directamente del campo {{Nombre_Campo}}. En este contexto no es recomendable, pero es admitido. "CODIGO": Código de la <i>Cuenta Contable</i> a utilizar, debería usarse solo una vez y ser el último de los parámetros de la lista.
Valor devuelto	El valor (cadena) del primero de los parámetros que no devuelva una cadena vacía. Si todos los parámetros devuelven una cadena vacía se lanza una excepción.
Observaciones	Las referencias que no existan no producen un error, pero generan una advertencia que puede ser visualizada al terminar el proceso de integración. En caso de no se obtenga una

	<p>cuenta contable válida (existente) se genera un error y la regla no se ejecuta.</p> <p>El resto de las funciones (especiales y comunes) pueden hacer referencia a ésta cuenta contable (a partir de ahora referida como "<i>La cuenta Contable Previamente Seleccionada</i>") y sus propiedades por medio del objeto <code>loCuentaContable</code>.</p>
--	--

Función: <code>mObtenerCuentaGastos (Param1 [,paramN]*)</code> → CADENA	
Contexto	Solo es válida en el campo "Cuenta de Gastos" de la regla de integración.
Parámetros	<p>Uno o más de los siguientes:</p> <ul style="list-style-type: none"> • <code>Contable_NombreTabla[N]</code>: Referencia a la <i>Cuenta de Gastos</i> almacenada en la página N° "N" en el campo de información contable <code>{{Contable_NombreTabla}}</code>. • <code>Cuenta_Integracion["CODIGO"]</code>: Referencia a la <i>Cuenta de Gastos</i> registrada en la <i>Cuenta Contable</i> relacionada a la <i>Cuenta de Integración</i> de código "CODIGO". • <code>{{cuenta_gastos}}</code>: Campo del cual se tomará el código de la <i>Cuenta de Gastos</i>. En este contexto no es recomendable, pero es admitido. • <code>loCuentaContable.pcCuentaGasto</code>: Indica que se tomará el código de la <i>Cuenta de Gastos</i> de la <i>Cuenta Contable</i> previamente seleccionada. • "CODIGO": Código de la <i>Cuenta de Gastos</i> a utilizar, debería usarse solo una vez y ser el último de los parámetros de la lista.
Valor devuelto	El valor (cadena) del primero de los parámetros que no devuelva una cadena vacía. Si todos los parámetros devuelven una cadena vacía se lanza una excepción.
Observaciones	Las referencias que no existan no producen un error, pero generan una advertencia que puede ser visualizada al terminar el proceso de integración. Si la <i>Cuenta Contable</i> previamente seleccionada maneja <i>Cuenta de Gastos</i> y no se obtiene una <i>Cuenta de Gastos</i> válida (existente) se genera un error y la integración se detiene. Si la <i>Cuenta Contable</i> previamente seleccionada no maneja <i>Cuenta de Gastos</i> ésta función nunca se evalúa y se deja en blanco el campo correspondiente en el renglón del comprobante.

Función: <code>mObtenerCentrosCostos (Param1 [,paramN]*)</code> → CADENA	
Contexto	Solo es válida en el campo "Centros de Costos" de la regla de integración.
Parámetros	<p>Uno o más de los siguientes:</p> <ul style="list-style-type: none"> • <code>Contable_NombreTabla[N]</code>: Referencia a la <i>Distribución por Centros de Costo</i> almacenada en la página N° "N" en el campo de información contable <code>{{Contable_NombreTabla}}</code>. • <code>Cuenta_Integracion["CODIGO"]</code>: Hace referencia a la <i>Distribución por Centros de Costo</i> (100% en un único centro de costos) registrada en la <i>Cuenta Contable</i> relacionada a la <i>Cuenta de Integración</i> de código "CODIGO". • <code>{{centro_costos}}</code>: Campo del cual se tomará la <i>Distribución por Centros de Costo</i>. En este contexto no es recomendable, pero es admitido. • <code>loCuentaContable.pcCentrosCostos</code>: Hace referencia a la <i>Distribución por Centros de Costo</i> de la <i>Cuenta Contable</i> previamente seleccionada. • "DISTRIBUCION": <i>Distribución por Centros de Costo</i> a utilizar, debería usarse solo una vez y ser el último de los parámetros de la lista (ver observaciones).

Valor devuelto	El valor (cadena) del primero de los parámetros que no devuelva una cadena vacía. Si todos los parámetros devuelven una cadena vacía se lanza una excepción.
Observaciones	<p>Las referencias que no existan no producen un error, pero generan una advertencia que puede ser visualizada al terminar el proceso de integración. Si no se obtiene una <i>Distribución por Centros de Costo</i> válida (todos los centros de costo deben existir y sumar 100%) se genera un error y la integración se detiene. El formato de una <i>Distribución por Centros de Costo</i> es: "CENTRO1[:dd.dd[;CENTRO2[:dd.dd]]*]", E.G.</p> <ul style="list-style-type: none"> "C-Ventas": 100% a la cuenta de costos C-Ventas. "C-Compras:80.00;Fletes:20": 80% a la cuenta de costos C-Compras y 20% a Fletes. "venValencia:45;venMaracay:37.8; venOtros": 45% a la cuenta de costos venValencia, 37.8% a venMaracay y el resto (17.2%) a venOtros.

Función: <code>mObtenerAuxiliar(Param1[:paramN]*)</code> → CADENA	
Contexto	Solo es válida en el campo "Auxiliar" de la regla de integración.
Parámetros	<p>Uno o más de los siguientes:</p> <ul style="list-style-type: none"> <code>Cuenta_Integracion["CODIGO"]</code>: Referencia al <i>Auxiliar</i> registrada en la <i>Cuenta Contable</i> relacionada a la <i>Cuenta de Integración</i> de código "CODIGO". <code>{{auxiliar}}</code>: Campo del cual se tomará el código de la <i>Auxiliar</i>. <code>oCuentaContable.Auxiliar</code>: Hace referencia al código del <i>Auxiliar</i> de la <i>Cuenta Contable</i> previamente seleccionada. "CODIGO": Código del <i>Auxiliar</i> a utilizar, debería usarse solo una vez y ser el último de los parámetros de la lista.
Valor devuelto	El valor (cadena) del primero de los parámetros que no devuelva una cadena vacía. Si todos los parámetros devuelven una cadena vacía el resultado devuelve una cadena vacía.
Observaciones	Las referencias que no existan no producen un error, pero generan una advertencia que puede ser visualizada al terminar el proceso de integración. Si la <i>Cuenta Contable</i> previamente seleccionada maneja <i>Auxiliar</i> y no se obtiene un <i>Auxiliar</i> válido (existente) se genera un error y la integración se detiene. Si la <i>Cuenta Contable</i> previamente seleccionada no maneja <i>Auxiliar</i> ésta función nunca se evalúa y se deja en blanco el campo correspondiente en el renglón del comprobante.

Funciones Comunes

Funciones Numéricas

Maximo (**vParametro1**; **vParametro2** [**;** **vParametroN**]*) → VALOR

Descripción: Devuelve el mayor de los parámetros. Se asume que todos los parámetros (y el resultado) son de mismo tipo que el primero de ellos. Ej:

Maximo (4.1; 3; -2) = 4.1

Maximo ("alfa"; "gamma"; "beta") = "gamma"

Maximo (#20081031#; #20081225#) = #20081225#

Parámetros:

1. **vParametro1**, **vParametro2...**: Dos o más valores numéricos, de fecha, de cadena, lógicos, NaN o Null.

Minimo (**vParametro1**; **vParametro2** [**;** **vParametroN**]*) → VALOR

Descripción: Devuelve el menor de los parámetros. Se asume que todos los parámetros (y el resultado) son de mismo tipo que el primero de ellos. Ej:

Minimo (4.1; 3; -2) = -2

Minimo ("alfa"; "gamma"; "beta") = "alfa"

Minimo (#20081031#; #20081225#) = #20081031#

Parámetros:

1. **vParametro1**, **vParametro2...**: Dos o más valores numéricos, de fecha, de cadena, lógicos, NaN o Null.

Redondear (**nNumero**; **nDecimales**) → NUMERO

Descripción: Devuelve el mismo número, redondeado al número de decimales indicado. El redondeo se hace hacia abajo (hacia "adentro" o hacia el 0) si el siguiente decimal es menor que 5, y hacia arriba ("afuera") si es igual o mayor a 5. Ej:

Redondear (4.125; 2) = 4.13

Redondear (227.20; -1) = 230

Parámetros:

1. **nNumero**: Valor numérico a ser redondeado.
2. **nDecimales**: Número entero de dígitos significativos a la derecha del punto decimal. Debe ser un entero entre -16 y 28 (la parte fraccional de **nDecimales** es truncada).

EnteroSuperior (**nNumero**) → NUMERO

Descripción: Devuelve el mismo número, redondeado al entero superior (hacia "la derecha"). Ej:

EnteroSuperior (4) = 4

EnteroSuperior (4.125) = 5

EnteroSuperior (-4.125) = -4

Parámetros:

1. **nNumero**: Valor numérico a ser redondeado.

EnteroInferior (**nNumero**) → NUMERO

Descripción: Devuelve el mismo número, redondeado al entero inferior (hacia "la izquierda"). Ej:

EnteroInferior (4) = 4

EnteroInferior (4.125) = 4

Parámetros:

1. **nNumero**: Valor numérico a ser redondeado.

EnteroInferior(-4.125) = -5	
Truncar (nNumero) → NUMERO	
<p>Descripción: Devuelve el mismo número sin parte decimal. Ej:</p> <p>Truncar(4) = 4</p> <p>Truncar(4.125) = 4</p> <p>Truncar(-4.125) = -4</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. nNumero: Valor numérico a ser redondeado.

Funciones de Cadena	
Longitud (cCadena) → NUMERO	
<p>Descripción: Devuelve la longitud de la cadena especificada.</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena de la que se obtendrá la longitud.
Recortar (cCadena) → CADENA	
<p>Descripción: Devuelve la cadena sin los espacios en blanco en los extremos.</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena cuyos espacios extremos serán eliminados.
TextoIzquierda (cCadena; nCantidad) → CADENA	
<p>Descripción: Devuelve una subcadena con el número especificado de caracteres contados desde el principio de la cadena original.</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena de la que se obtendrá la subcadena. 2. nCantidad: Cantidad de caracteres que serán devueltos.
TextoDerecha (cCadena; nCantidad) → CADENA	
<p>Descripción: Devuelve una subcadena con el número especificado de caracteres contados desde el final de la cadena original.</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena de la que se obtendrá la subcadena. 2. nCantidad: Cantidad de caracteres que serán devueltos.
TextoCentro (cCadena; nDesde [; nCantidad]) → CADENA	
<p>Descripción: Devuelve una subcadena con el número especificado de caracteres contados desde la posición indicada de la cadena original. Ej:</p> <p>TextoCentro("Comentario"; 4; 3) = "ent"</p> <p>TextoCentro("Comentario"; 4) = "entario"</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena de la que se obtendrá la subcadena. 2. nDesde: Posición desde la cual se obtendrá la subcadena. 3. nCantidad: Cantidad de caracteres que serán devueltos. Si no se indica una cantidad devuelve todos los caracteres a partir de nDesde.
NoVacio (cCadena1; cCadena2 [; cCadenaN]*) → CADENA	
<p>Descripción: Devuelve la primera cadena pasada como parámetro que no esté vacía; si todas las cadenas están vacías devuelve una cadena vacía. Ej.</p>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena1, cCadena2...: Dos o más cadenas a evaluar.

<pre>NoVacio(""); "q"; " ") = "q" NoVacio(""); " "; "q") = " "</pre>	
EsVacio(cCadena) → Logico	
<p>Descripción: Devuelve un valor lógico que indica si la cadena indicada está o no vacía. Solo se considera vacía a una cadena de longitud cero. Ej.</p> <pre>EsVacio("texto de prueba") = false EsVacio("") = true EsVacio(" ") = false</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. cCadena: Cadena a evaluar.

Funciones de Fecha	
Hoy() → FECHA	
<p>Descripción: Devuelve la fecha actual, con las componentes de hora establecidas a 0. Ej.</p> <pre>Hoy() = #20090115#</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. Ninguno.
Ahora() → FECHA	
<p>Descripción: Devuelve la fecha y hora actual. Ej.</p> <pre>Ahora() = #20090115 14:25:22.159#</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. Ninguno.
NuevaFecha(nAño; nMes; nDia[; nHoras[; nMinutos[; nSegundos[; nMilisegundos]]]) → FECHA	
<p>Descripción: Devuelve un valor de fecha creado a partir de las componentes indicadas. Los valores de las componentes son corregidos en caso de salir del rango de valores válidos. Ej.</p> <pre>NuevaFecha(2009; 1; 15) = #20090115# NuevaFecha(2009; 1; 15; 14) = #20090115 14:00# NuevaFecha(2009; 1; 15; 14; 25; 10; 22) = #20090115 14:25:10.022#</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. nAño, nMes, nDia: Parámetros enteros obligatorios; la parte fraccional es truncada. 2. nHoras, nMinutos, nSegundos, nMilisegundos: Parámetros enteros opcionales; la parte fraccional es truncada.
ComponenteFecha(dFecha; cUnidad) → NUMERO	
<p>Descripción: Devuelve el componente cUnidad de la fecha indicada. El día de la semana devuelve 0 para domingo, 1 para lunes... y 6 para sábado.</p> <pre>ComponenteFecha(#20090215#; "AÑO") = 2 ComponenteFecha(#20090215#; "DIA") = 15 ComponenteFecha(#20090215#; "DIA-SEMANA") = 0 ComponenteFecha(#20090215 12:35:12#; "MINUTO") = 35</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> 1. dFecha: Fecha de la que se obtendrá el componente. 2. cUnidad: Uno de los siguientes valores de cadena: "SEGUNDO", "MINUTO", "HORA", "DIA", "DIA-SEMANA", "DIA-AÑO", "MES", "AÑO".
DiferenciaFechas(dFechaInicial; dFechaFinal [; cUnidad]) → FECHA	

<p>Descripción: Devuelve la cantidad de unidades de tiempo (enteras) desde dFechaInicial hasta dFechaFinal; las unidades de tiempo menores a cUnidad <i>son tomadas en cuenta</i> en el conteo (para evitar esto deben estar redondeadas a la unidad deseada). Si dFechaInicial es mayor que dFechaFinal el resultado es negativo. Ej:</p> <pre>DiferenciaFechas(#20081220#; #20081230#)=10 DiferenciaFechas(#20081220 23:59#; #20081230 00:00#)=10 DiferenciaFechas(#20090115#; #20081230#)=-16</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFechaInicial: Fecha desde la cual se contará. dFechaFinal: Fecha hasta la cual se contará. cUnidad: Uno de los siguientes valores de cadena: "SEGUNDO" , "MINUTO" , "HORA" , "DIA" , "MES" , "AÑO". Si se omite se usará "DIA".
<p>TruncarFecha (dFecha) → FECHA</p>	
<p>Descripción: Devuelve la misma fecha con las componentes de hora establecidas a 0. Ej:</p> <pre>TruncarFecha(#20081220 15:00:25#) = #20081220# TruncarFecha(#20081220 00:00:00#) = #20081220#</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFecha: Fecha a la cual se eliminará (o establecerá a cero) las componentes de hora.
<p>SumarComponenteFecha (dFecha; dCantidad; cUnidad) → FECHA</p>	
<p>Descripción: Devuelve la fecha dFecha luego de sumarle dCantidad unidades de tiempo de tipo cUnidad. La suma se realiza considerando los años bisiestos y el numero de días máximo de cada mes: Si al sumar meses la componente de día no es válida, ésta es ajustada a una fecha válida (por ejemplo, al sumar un mes al 31 de Enero se obtiene 28 de Febrero). Ej:</p> <pre>SumarComponenteFecha(#20081231#; 1; "DIA") = #20090101# SumarComponenteFecha(#20081231#; 1.75; "MINUTO") = #20081231 00:01:45# SumarComponenteFecha(#20090131 14:25:05#; -3; "MES") = #20081031 14:25:05# SumarComponenteFecha(#20090131 14:25:05#; 1; "MES") = #20090228 14:25:05#</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFecha: A la cual se le sumará (o restará) el número de unidades indicadas. dCantidad: Cantidad de unidades que se sumará o restará a dFecha. Si cUnidad es "MES" o "AÑO" se truncará la parte decimal de dCantidad, en caso contrario la parte decimal se agregará coherentemente a las unidades menores cUnidad: Uno de los siguientes valores de cadena: "SEGUNDO" , "MINUTO" , "HORA" , "DIA" , "MES" , "AÑO".
<p>ObtenerAño (dFecha) → NUMERO</p>	
<p>Descripción: Devuelve el valor del año de la fecha indicada. Ej:</p> <pre>ObtenerAño(#20081220 15:00:25#) = 2008</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFecha: Fecha de la cual se obtendrá el valor del año.
<p>ObtenerMes (dFecha) → NUMERO</p>	
<p>Descripción: Devuelve el valor del mes de la fecha indicada. Ej:</p> <pre>ObtenerMes(#20081220 15:00:25#) = 12</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFecha: Fecha de la cual se obtendrá el valor del año.
<p>ObtenerDia (dFecha) → NUMERO</p>	
<p>Descripción: Devuelve el valor del día del mes de la fecha indicada. Ej:</p> <pre>ObtenerDia(#20081220 15:00:25#) = 20</pre>	<p>Parámetros:</p> <ol style="list-style-type: none"> dFecha: Fecha de la cual se obtendrá el valor del día del mes.

