

Funciones Lógicas

Las funciones lógicas devuelven verdadero o falso en forma de 1 y 0, por lo que las casillas de verificación pueden rellenarse con estas funciones. También se pueden usar como la condición lógica para la función ifThenElse.

igual

Firma	{igual (parámetro1; parámetro2)}
Parámetros	parámetro1: puede ser cualquier valor de cualquier tipo
	parámetro2: puede ser cualquier valor de cualquier tipo
Descripción	Determina si el parámetro1 es igual con parameter2
Devoluciones	1 si los dos parámetros son iguales o 0 si no
Ejemplo de llamada	{igual (1; 2)} devuelve 0

no es igual

Firma	<code>{notEqual (parámetro1; parámetro2)}</code>
Parámetros	parámetro1: puede ser cualquier valor de cualquier tipo
	parámetro2: puede ser cualquier valor de cualquier tipo
Descripción	Determina si el parámetro1 no es igual a parameter2
Devoluciones	0 si los dos parámetros son iguales o 1 si no
Ejemplo de llamada	<code>{notEqual (1; 2)}</code> devuelve 1

mas grande que

Firma	<code>{greaterThan (parameter1; parameter2)}</code>
Parámetros	parámetro1: puede ser cualquier valor de cualquier tipo

	<p>parámetro2: puede ser cualquier valor de cualquier tipo</p>
Descripción	<p>Determina si el parámetro1 es mayor que parameter2</p>
Devoluciones	<p>1 si parámetro1 mayor que parameter2 , 0 si no</p>
Ejemplo de llamada	<p>{greaterThan (3; 3)} devuelve 0</p>

mayor que o igual

Firma	<pre>{greaterThanOrEqualTo (parameter1; parámetro2)}</pre>
Parámetros	<p>parámetro1: puede ser cualquier valor de cualquier tipo</p>
	<p>parámetro2: puede ser cualquier valor de cualquier tipo</p>
Descripción	<p>Determina si el parámetro1 es mayor</p>

	que o igual parámetro2
Devoluciones	1 si parámetro1 mayor que o igual parámetro2 , 0 no si
Ejemplo de llamada	{greaterThanOrEqualTo (3; 3)} devuelve 1

menos que

Firma	{menosThan (parámetro1; parámetro2)}
Parámetros	parámetro1: puede ser cualquier valor de cualquier tipo
	parámetro2: puede ser cualquier valor de cualquier tipo
Descripción	Determina si el parámetro 1 es menor que parámetro2
Devoluciones	1 si parámetro1 menos de parameter2 , 0 si no

Ejemplo de llamada	<code>{lessThan (3; 3)}</code> devuelve 0
--------------------	---

menor o igual

Firma	<code>{lessThanOrEqualTo (parámetro1; parámetro2)}</code>
Parámetros	parámetro1: puede ser cualquier valor de cualquier tipo
	parámetro2: puede ser cualquier valor de cualquier tipo
Descripción	Determina si el parámetro 1 es menor que o parámetro igual2
Devoluciones	1 si parámetro1 menor que o igual parameter2 , 0 si no
Ejemplo de llamada	<code>{lessThanOrEqualTo (3; 3)}</code> devuelve 1

vacío

Firma	{vacío (parámetro)}
Parámetros	parámetro: valor del texto
Descripción	Determina si el parámetro está vacío
Devoluciones	1 si el parámetro está vacío, 0 si no
Ejemplo de llamada	{empty (cualquier texto)} devuelve 0

no vacío

Firma	{notEmpty (parámetro)}
Parámetros	parámetro: valor del texto
Descripción	Determina si el parámetro no es vacío
Devoluciones	1 si el parámetro no está vacío, 0 si vacío
Ejemplo de llamada	{notEmpty (cualquier texto)} devuelve 1

no

Firma	{no (parámetro)}
Parámetros	parámetro: valor lógico
Descripción	Niega el valor lógico de la parámetro
Devoluciones	1 si el parámetro es 0, 0 si parámetro es 1
Ejemplo de llamada	{not (0)} devuelve 1

y

Firma	{y (parámetro1; parámetro2)}
Parámetros	parámetro1: valor lógico
	parámetro2: valor lógico
Descripción	Aplica el operador lógico AND a dos

	valores lógicos
Devoluciones	1 si parámetro1 y parámetro2 es 1, 0 si alguno de los parámetros es 0
Ejemplo de llamada	{y (1; 0)} devuelve 0

o

Firma	{o (parámetro1; parámetro2)}
Parámetros	parámetro1: valor lógico
	parámetro2: valor lógico
Descripción	Aplica el operador lógico OR a dos valores lógicos
Devoluciones	1 si parámetro1 o parámetro2 es 1, 0 si ambos parámetros son 0
Ejemplo de llamada	{o (1; 0)} devuelve 1

Funciones de texto

Las funciones de texto se utilizan para manipular texto de varias maneras. Todas las funciones enumeradas aquí son totalmente compatibles con textos UTF-8, por lo que los caracteres especiales no deberían plantear ningún problema.

subcadena

Firma	{subcadena (texto; inicio; longitud)}
Parámetros	texto: valor de texto
	inicio: valor decimal
	longitud [parámetro opcional]: decimal valor
Descripción	Corta la subcadena de un campo de texto desde el inicio . Si no se establece el parámetro opcional de longitud , corta todos los caracteres hasta el final de la cadena, de lo contrario corta la longitud proporcionada. Indexación de los caracteres de un texto a partir de 0.
Devoluciones	Subcadena del texto dado

Ejemplo de llamada	<code>{substring (Este es mi texto; 5)}</code> devuelve es mi texto
	<code>{subcadena (Este es mi texto; 5; 5)}</code> vuelve es mi

longitud

Firma	<code>{longitud (parámetro)}</code>
Parámetros	parámetro: valor del texto
Descripción	Cuenta los personajes en un texto.
Devoluciones	El recuento de los personajes en un texto.
Ejemplo de llamada	<code>{longitud (texto de muestra)}</code> devuelve 11

reemplazar

Firma	<code>{reemplazar (buscar; reemplazar; asunto)}</code>
Parámetros	búsqueda: valor de texto

	reemplazar: valor de texto
	asunto: valor del texto
Descripción	Reemplazar todas las apariciones de búsqueda para reemplazar en el tema de texto .
Devoluciones	sujeto con valores reemplazados.
Ejemplo de llamada	{replace (apple; orange; Esto es una manzana árbol)} devuelve Esto es un árbol de naranja

posición

Firma	{posición (asunto, búsqueda)}
Parámetros	asunto: valor del texto
	búsqueda: valor de texto
Descripción	Encontrar la posición de la primera aparición de

	buscar en un tema
Devoluciones	Posición numérica de búsqueda en sujeto o -1 si la búsqueda no está presente en el asunto
Ejemplo de llamada	{posición (¿Dónde está mi texto ?; texto)} regresa 12

minúscula

Firma	{minúsculas (parámetro)}
Parámetros	parámetro: valor del texto
Descripción	Hacer texto en minúsculas
Devoluciones	El texto en minúscula
Ejemplo de llamada	{minúsculas (ESTO ES UN TEXTO SIMPLE)} devuelve esto es un texto de muestra

mayúscula

Firma	{mayúscula (parámetro)}
Parámetros	parámetro: valor del texto
Descripción	Hacer texto en mayúscula
Devoluciones	El texto en mayúscula
Ejemplo de llamada	{mayúscula (ESTO ES UN TEXTO SENCILLO)} devuelve ESTE ES UN TEXTO MUESTRA

Funciones matemáticas

Las funciones matemáticas se usan para manipular números de varias maneras. Varios operadores matemáticos se implementan como funciones en Calculate Fields.

añadir

Firma	{add (parámetro1; parámetro2)}
Parámetros	parámetro1: valor numérico
	parámetro2: valor numérico

Descripción	Agrega parameter1 y parameter2
Devoluciones	La suma de parameter1 y parámetro2
Ejemplo de llamada	{add (3.12; 4.83)} devuelve 7.95

sustraer

Firma	{restar (parámetro1; parámetro2)}
Parámetros	parámetro1: valor numérico
	parámetro2: valor numérico
Descripción	Resta el parámetro2 de parámetro1
Devoluciones	La distinción de parameter2 y parámetro1
Ejemplo de llamada	{restar (8; 3)} devuelve 5

multiplicar

Firma	{multiplicar (parámetro1; parámetro2)}
Parámetros	parámetro1: valor numérico
	parámetro2: valor numérico
Descripción	Multiplica parámetro1 y parámetro2
Devoluciones	El producto de parameter1 y parámetro2
Ejemplo de llamada	{multiplicar (2; 4)} devuelve 8

divide

Firma	{divide (parámetro1; parámetro2)}
Parámetros	parámetro1: valor numérico
	parámetro2: valor numérico

Descripción	Divide el parámetro2 con el parámetro1
Devoluciones	La división de parameter2 y parámetro1
Ejemplo de llamada	{divide (8; 2)} devuelve 4

poder

Firma	{potencia (parámetro1; parámetro2)}
Parámetros	parámetro1: valor numérico
	parámetro2: valor numérico
Descripción	Eleva el parámetro1 a la potencia de parámetro2
Devoluciones	parámetro1 elevado a la potencia de parámetro2
Ejemplo de llamada	{power (2; 7)} devuelve 128

raíz cuadrada

Firma	{squareRoot (parámetro)}
Parámetros	parámetro: valor numérico
Descripción	Calcula la raíz cuadrada del parámetro
Devoluciones	La raíz cuadrada del parámetro
Ejemplo de llamada	{squareRoot (4)} devuelve 2

absoluto

Firma	{Absolute (parámetro)}
Parámetros	parámetro: valor numérico
Descripción	Calcula el valor absoluto de parámetro
Devoluciones	El valor absoluto del parámetro

Ejemplo de llamada	{absolute (-4)} devuelve 4
--------------------	----------------------------

Funciones de fecha

Hay varias funciones de fecha implementadas en Calculate Fields, por lo que el usuario puede manipular las fechas de muchas maneras. La mayoría de las funciones usa un parámetro de formato, que se utiliza para establecer el resultado de las funciones formateadas como desea el usuario. Las opciones para estos formatos son equivalentes con los parámetros de formato PHP:

Por día		
re	Día del mes, 2 dígitos con ceros	01 a 31
re	Una representación textual de un día, tres letras	De lunes a domingo
j	Día del mes sin ceros a la izquierda	1 a 31

I	Una representación textual completa del día de la semana	Domingo a Sábado
norte	Representación numérica ISO-8601 de día de la semana	1 (para el lunes) a 7 (para el domingo)
S	Sufijo ordinal Inglés para el día de la semana, 2 caracteres	st, nd, rd o th. Funciona bien con j
en	Representación numérica del día de la semana	0 (para el domingo) a 6 (para el sábado)
de	El día del año (empezando desde 0)	0 a 365
Por semana		
la	ISO-8601 semana número de años, semanas comenzando el lunes	42 (la 42 ^a semana del año)

Por mes		
F	Una representación textual completa de un mes, como enero o marzo	Enero a Diciembre
metro	Representación numérica de un mes, con ceros a la izquierda	01 a 12
METRO	Una breve representación textual de un mes, tres letras	De enero a diciembre
norte	Representación numérica de un mes, sin ceros a la izquierda	1 a 12
t	Número de días en el mes dado	28 a 31
Por año		

la	Si es un año bisiesto	1 si es un año bisiesto, 0 de lo contrario
o	Número de año ISO-8601. Esto tiene el mismo valor como Y, excepto que si el número de semana ISO (W) pertenece al año anterior o al siguiente, ese año se usa en su lugar	1999 o 2003
Y	Una representación numérica completa de un año, 4 dígitos	1999 o 2003
y	Una representación de dos dígitos de un año	99 o 03
Para el tiempo		
a	Antes del mediodía y por la tarde en minúsculas	am o PM
A	Antes del mediodía y por la tarde en mayúsculas	Am o PM

B	Swatch horario de Internet	000 a 999
gramo	Formato de 12 horas de una hora sin liderazgo ceros	1 a 12
GRAMO	Formato de 24 horas de una hora sin ventaja ceros	0 a 23
h	Formato de 12 horas de una hora con liderazgo ceros	01 a 12
H	Formato de 24 horas de una hora con liderazgo ceros	00 a 23
yo	Minutos con ceros a la izquierda	00 a 59
s	Segundos, con ceros a la izquierda	00 a 59
Para la zona horaria		

y	Identificador de zona horaria	UTC, GMT, Atlántico / Azores
I	Si la fecha está o no en la luz del día ahorrando tiempo	1 si Horario de verano, 0 de lo contrario
O	Diferencia con la hora de Greenwich (GMT) en horas	+0200
PAG	Diferencia con la hora de Greenwich (GMT) con dos puntos entre horas y minutos	+02:00
T	Abreviación de zona horaria	EST, MDT
con	Timezone offset en segundos. El desplazamiento para las zonas horarias al oeste de UTC siempre es negativo, y para aquellos al este de UTC siempre es positivo.	-43200 a 50400
Para fecha / hora completa		

c	Fecha ISO 8601	2004-02-12T15: 19: 21 + 00: 00
r	Fecha de formato RFC 2822	Jue, 21 de diciembre de 2000 16:01:07 +0200
U	Segundos desde la Época de Unix (1 de enero 1970 00:00:00 GMT)	

Para todas las funciones sin parámetro de marca de tiempo,
suponemos que la fecha / hora actual es 2016.04.29. 15:08:03

fecha

Firma	{date (format; timestamp)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
Descripción	Crea una fecha en el formato dado

Devoluciones	marca de tiempo en el formato dado
Ejemplo de llamada	{date (ymd; 2016-02-11)} devuelve 160211

ahora

Firma	{ahora (formato)}
Parámetros	formato: texto de formato
Descripción	Crea la fecha / hora real en el dado formato
Devoluciones	Fecha / hora actual en el formato
Ejemplo de llamada	{ahora (Ymd H: i: s)} regresa 2016-04-29 15:08:03

ayer

Firma	{ayer (formato)}
-------	------------------

Parámetros	formato: texto de formato
Descripción	Crea la fecha / hora de ayer en el formato dado
Devoluciones	La fecha / hora de ayer en el formato dado
Ejemplo de llamada	{ayer (Ymd H: i: s)} devuelve 2016-04-28 15:08:03

mañana

Firma	{mañana (formato)}
Parámetros	formato: texto de formato
Descripción	Crea la fecha / hora de mañana en el formato dado
Devoluciones	La fecha / hora de mañana en el formato dado

Ejemplo de llamada	<code>{tomorrow (Ymd H: i: s)}</code> devuelve 2016-04-30 15:08:03
--------------------	---

datediff

Firma	<code>{datediff (timestamp1; timestamp2; unit)}</code>
Parámetros	timestamp1: valor de fecha / hora
	timestamp2: valor de fecha / hora
	unidad: años / meses / días / horas / minutos / segundos; predeterminado: días
Descripción	Resta la marca de tiempo2 de timestamp1
Devoluciones	La diferencia entre las dos fechas devuelto en unidad
Ejemplo de llamada	<code>{datediff (2016-02-01; 2016-04-22; días)}</code>

	devuelve 81
--	-------------

addYears

Firma	{addYears (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Añade cantidad años a fecha y hora
Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	{addYears (Ymd; 2016-04-22; 1)} devuelve 20170422

addMonths

Firma	{addMonths (formato; marca de tiempo; cantidad)}
-------	--

Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Agrega una cantidad de meses a la marca de tiempo
Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	<pre>{addMonths (Ymd; 2016-04-22; 1)}</pre> devoluciones 20160522

addDays

Firma	<pre>{addDays (formato; marca de tiempo; cantidad)}</pre>
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora

	cantidad: número decimal
Descripción	Agrega días de cantidad a la marca de tiempo
Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	{addDays (Ymd; 2016-04-22; 1)} devoluciones 20160423

addHours

Firma	{addHours (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Agrega horas de cantidad a la marca de tiempo

Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	<code>{addHours (Ymd H: i: s; 2016-04-22 23:30; 5)}</code> devuelve 20160423 04:30:00

addMinutes

Firma	<code>{addMinutes (formato; marca de tiempo; cantidad)}</code>
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Agrega minutos de cantidad a la marca de tiempo
Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	<code>{addMinutes (Ymd H: i: s; 2016-04-22 22:58; 5)}</code> devuelve 20160422 23:03:00

addSeconds

Firma	{addSeconds (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Agrega cantidad de segundos a la marca de tiempo
Devoluciones	Fecha incrementada en formato
Ejemplo de llamada	{addSeconds (Ymd H: i: s; 2016-04-22 22:58; 5)} devuelve 20160422 22:58:05

restar Año

Firma	{restarAños (formato; marca de tiempo; cantidad)}
-------	---

Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Resta restan años desde marca de tiempo
Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{restar años (Ymd; 2016-04-22; 5)} devuelve 20110422

restaMedios

Firma	{subtractMonths (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora

	cantidad: número decimal
Descripción	Los restas suman meses desde marca de tiempo
Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{restarMes (Ymd; 2016-04-22; 5)} vuelve 20151122

restarDías

Firma	{restractDays (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Resta resta cantidad de días desde marca de tiempo

Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{retractDays (Ymd; 2016-04-22; 5)} devuelve 20160417

restar horas

Firma	{restar horas (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Los restas montan horas desde marca de tiempo
Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{restar horas (Ymd H: i: s; 2016-04-22 12:37; 5)} devuelve 20160422 07:37:00

restar minutos

Firma	{restar minutos (formato; marca de tiempo; cantidad)}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Resta cantidad minutos de marca de tiempo
Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{restar minutos (Ymd H: i: s; 2016-04-22 12:37; 5)} devuelve 20160422 12:32:00

restar segundos

Firma	{subtractSeconds (formato; marca de tiempo;
-------	---

	cantidad}}
Parámetros	formato: texto de formato
	marca de tiempo: valor de fecha / hora
	cantidad: número decimal
Descripción	Resta cantidad minutos de marca de tiempo
Devoluciones	Fecha decrementada en formato
Ejemplo de llamada	{restarSeconds (Ymd H: i: s; 2016-04-22 12:37; 5)} devuelve 20160422 12:36:55

Funciones de control

Solo hay una función de control implementada en Calculate Fields hasta el momento, pero esta función garantiza que el usuario pueda escribir fórmulas muy complejas con condiciones. Dado que las funciones pueden integrarse entre sí, el usuario puede escribir uniones con muchas ramas.

ifThenElse

Firma	<code>{ifThenElse (condición; trueBranch; falseBranch)}</code>
Parámetros	condición: valor lógico
	trueBranch: cualquier expresión
	falseBranch: cualquier expresión
Descripción	Selecciona una de las dos ramas dependiendo en la condición
Devoluciones	trueBranch si la condición es true, falseBranch de lo contrario
Ejemplo de llamada	<code>{ifThenElse ({igual (1; 1)}; 1 es igual a 1; 1 no es igual a 1)}</code> devuelve 1 es igual a 1

Contadores

Hay varios contadores implementados en Calculate Fields que se pueden usar en varios escenarios.

Los contadores ordenados en dos grupos:

Contadores globales: contadores que se incrementan cada vez que se evalúa una fórmula afectada

Contadores diarios: contadores que se restablece todos los días. (A partir de 1)

En este capítulo suponemos que el valor actual de los contadores es 4, por lo que el valor incrementado será 5 con el formato dado.

GlobalCounter

Firma	{GlobalCounter (nombre; numberLength)}
Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombrar con longitud numberLength
Devoluciones	Contador con longitud numberLength
Ejemplo de llamada	{GlobalCounter (myName; 4)} devuelve 0005

GlobalCounterPerUser

Firma	{GlobalCounterPerUser (nombre;
-------	--------------------------------

	<code>numberLength}}</code>
Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombre para el usuario que crea la entidad con el número de longitud Longitud
Devoluciones	Contador con longitud numberLength
Ejemplo de llamada	<code>{GlobalCounterPerUser (myName; 3)}</code> returns 005

GlobalCounterPerModule

Firma	<code>{GlobalCounterPerModule (nombre; numberLength)}</code>
Parámetros	nombre: cualquier texto
	numberLength: número decimal

Descripción	Incrementa y devuelve el contador para nombre para el módulo de la entidad con el número de longitud Longitud
Devoluciones	Contador con longitud numberLength
Ejemplo de llamada	{GlobalCounterPerModule (myName; 2)} vuelve 05

GlobalCounterPerUserPerModule

Firma	{GlobalCounterPerUserPerModule (nombre; numberLength)}
Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombre para el usuario que crea la entidad y para el módulo de la entidad con el número de longitud Longitud

Devoluciones	Contador con longitud numberLength
Ejemplo de llamada	{GlobalCounterPerUserPerModule (myName; 1)} devuelve 5

DailyCounter

Firma	{DailyCounter (nombre; numberLength)}
Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombrar con longitud numberLength
Devoluciones	Contador con el número de longitud Longitud , o si el contador no se incrementa este día, entonces 1 con el número de longitud Largo
Ejemplo de llamada	{DailyCounter (myName; 1)} devuelve 5

DailyCounterPerUser

Firma	{DailyCounterPerUser (nombre; numberLength)}
Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombre para el usuario que crea la entidad con el número de longitud Longitud
Devoluciones	Contador con el número de longitud Longitud , o si el contador no se incrementa este día para este usuario, entonces 1 con el número de longitud Longitud
Ejemplo de llamada	{DailyCounter (myName; 1)} devuelve 5

DailyCounterPerModule

Firma	{DailyCounterPerModule (nombre; numberLength)}
-------	--

Parámetros	nombre: cualquier texto
	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombre para el módulo de la entidad con el número de longitud Longitud
Devoluciones	Contador con el número de longitud Longitud , o si el contador no se incrementa este día para este módulo, entonces 1 con el número de longitud Largo
Ejemplo de llamada	<code>{DailyCounterPerModule (myName; 1)}</code> devuelve 5

DailyCounterPerUserPerModule

Firma	<code>{DailyCounterPerUserPerModule (name; numberLength)}</code>
Parámetros	nombre: cualquier texto

	numberLength: número decimal
Descripción	Incrementa y devuelve el contador para nombre para el usuario que crea la entidad y para el módulo de la entidad con el número de longitud Longitud
Devoluciones	Contador con el número de longitud Longitud , o si el contador no se incrementa este día para el usuario que crea la entidad y para este módulo, entonces 1 con el número de longitud Longitud
Ejemplo de llamada	{DailyCounterPerUserPerModule (myName; 1)} devuelve 5

Ejemplo

Calcule la tarifa mensual por una oportunidad

Caso de uso

El usuario desea calcular una tarifa mensual de una oportunidad para un campo personalizado dividiendo la cantidad de la oportunidad por la duración.

Preparar

Nuestro módulo de oportunidades tiene un campo desplegable llamado Duración con valores: (valor de la base de datos entre paréntesis) 6 meses [6], 1 año [12], 2 años [24]. También hay un campo de moneda llamado Mensual.

Flujo de trabajo

Vaya al módulo Flujo de trabajo y cree un nuevo Flujo de trabajo. Establezca las opciones base como las siguientes:

Nombre: como quieras	Módulo Flujo de trabajo: Oportunidades
Estado: activo	Ejecutar: solo al guardar
Ejecutar encendido: todos los registros	Repeticiones repetidas: verificado

The screenshot shows a configuration form for a workflow. The fields are as follows:

- Name:** Monthly
- Assigned to:** Administrator
- Workflow Module:** Opportunities
- Status:** Active
- Run:** Only On Save
- Run On:** All Records
- Repeated Runs:**
- Description:** (Empty text area)

No creamos ninguna condición, ya que nos gustaría que el flujo de trabajo se ejecute en todas las oportunidades.

Ahora, agregue una acción y seleccione Calcular campos en el menú desplegable.

Luego, agregue dos campos de Oportunidades como parámetros. Primero, seleccione Cantidad de oportunidad (cantidad) y agréguela como parámetro (será {P0}) luego seleccione Duración y la opción de valor sin procesar del menú desplegable de tipo de datos y agréguela como parámetro dos (será {P1}). No es necesario agregar ningún parámetro relacional para esta fórmula.

Ahora, agregue una fórmula para el campo mensual y complete el cuadro de texto con la siguiente fórmula:

{divide ({P0}; {P1})}

Entonces toda la acción debería verse así:

The screenshot shows a workflow configuration window titled 'Calculate Fields'. It includes a 'Name' field, a 'Parameters' section with a table of field names, value types, and identifiers, a 'Relation parameters' section with a dropdown for 'Users : Created by User' and an 'ID' field, and a 'Formulas' section with a table for field names and formulas.

Field name	Value type	Identifier
Opportunity Amount	Raw value	{P0}
Duration	Raw value	{P1}

Relation parameters

Users : Created by User | ID

Field name	Formula
Monthly	{divide({P0}; {P1})}

Guarde WorkFlow y cree una nueva oportunidad:

[Save](#) [Cancel](#) [View Change Log](#)

Opportunity Name: *

Currency:

Opportunity Amount: *

Sales Stage: *

Probability (%):

Next Step:

Description:

Account Name: *

Expected Close Date: *

Type:

Lead Source:

Campaign:

Duration:

Assigned to:

^ Other

Como puede ver, ni siquiera agregamos el campo mensual a Editar Vista, porque no queremos obligar al usuario a realizar cálculos. Guarde la oportunidad y verifique los resultados en Vista Detallada:

My opportunity ☆ [Create](#)

Edit | (1 of 50) ▶

Basic	Other
Opportunity Name: My opportunity Opportunity Amount (USD \$): 1,830.00 Sales Stage: Prospecting Probability (%): 10 Next Step: Description: Assigned to: Administrator	Account Name: Super Star Holdings Inc Monthly: 152.50 Expected Close Date: 28.05.2016 Lead Source: Campaign: Duration: 1 year Type: