



Criterios, expresiones y funciones frecuentes

Empleo de expresiones, criterios y funciones frecuentes en consultas, formularios, informes y VBA. Especial funciones fecha

Jose Ignacio González Gómez

Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna

www.jggomez.eu

INDICE

1	Criterios y Expresiones frecuentes en Access	4
1.1	Introducción	4
1.2	Diseñando los criterios: la fila de criterios.....	4
1.2.1	Combinando criterios (Operador Y)	4
1.2.2	Criterios alternativos (Operador O).....	5
1.3	Principales tipos de criterios y expresiones empleadas.....	6
1.3.1	Expresiones para campos Texto, Memo e Hipervínculo. Ejemplos	6
1.3.2	Separar nombre y apellidos en una tabla de personas	8
1.3.3	Expresiones "Valores de texto, parciales y coincidentes". Ejemplos	10
1.3.4	Expresiones con "Parte de un valor de campo" (Como). Ejemplos	10
1.3.5	Expresiones. Manipular valores de texto.....	11
1.3.6	Expresiones para campo fecha/hora. Ejemplos	11
1.3.7	Expresiones con Rangos de Valores (>, <, >=, <=, <>, o Entre...Y). Ejemplos.....	14
1.3.8	Expresiones con Valores que no coinciden (Negado). Ejemplos	14
1.3.9	Expresiones con Valores de una lista (En). Ejemplos.....	14
1.3.10	Expresiones condicionales Si/Entonces. Ejemplos	15
1.3.11	Un valor de campo vacío (Nulo o cadena de longitud cero). Ejemplos	15
1.3.12	Expresiones para otros campos	16
2	Especial sobre el uso del carácter comodín	17
2.1	Introducción	17
2.2	Uso de carácter comodín en cuadro de diálogos Buscar y Reemplazar y Consultas de selección y actualización (ANSI-89).....	17

2.3	Uso de carácter comodín en cuadro de diálogos Buscar y Reemplazar y consultas de selección y actualización en proyectos Access (ANSI-92).	18
2.4	Tipos de datos que se pueden buscar mediante caracteres comodín	18
3	Especial: Operando con campo fecha - hora	19
3.1	Introducción, ¿Para qué las funciones fecha y hora?	19
3.2	Función DifFecha("aaaa", [FechaNacimiento], Fecha()) > 30	20
3.2.1	Objetivo de la función DifFecha o DateDiff	20
3.2.2	Sintaxis de la función	20
3.3	Formato fecha y hora	20
3.3.1	Introducción. Fecha hora como numero	20
3.3.2	Formatos de fecha y hora personalizados. Dia de la semana, del mes, semana del año, etc	21
3.4	Ejemplos de manipulación y cálculo con campo fechas	22
3.4.1	Ejemplo	22
3.4.2	Ejemplo	22
3.4.3	Ejemplo	23
3.4.4	Ejemplo	23
4	Empleo de funciones de agregado de dominio frecuentes	24
4.1	Introducción	24
4.2	Función DSuma ()	25
4.3	Función DlookUp ()	26
4.4	Función DCont ()	28
4.5	Función DBúsq ()	28
4.6	Funciones DMín (DMin) y DMáx (DMax)	28
4.7	Ejemplos expresiones que utilizan funciones de agregado de dominio	28
4.7.1	Ejemplos generales	28
4.7.2	Ejemplo. Resultado de una función de agregado de dominio	29
4.7.3	Ejemplo I. Crear una suma continua en un formulario. DSuma() con una condición	29
4.7.4	Ejemplo II. DSuma() con varias condiciones y de parámetro	29
5	Anexos y bibliografía	31
5.1	Principales funciones organizadas por categorías	31
5.1.1	ActiveX	31
5.1.2	Aplicación	31
5.1.3	Matrices	31
5.1.4	Conversión	31
5.1.5	Base de datos	31

5.1.6	<i>Fecha/Hora</i>	31
5.1.7	<i>Agregado de dominio</i>	32
5.1.8	<i>Tratamiento de errores</i>	32
5.1.9	<i>Entrada y salida de archivos</i>	32
5.1.10	<i>Administración de archivos</i>	32
5.1.11	<i>Financieras</i>	32
5.1.12	<i>Inspección</i>	33
5.1.13	<i>Matemáticas</i>	33
5.1.14	<i>Mensajes</i>	33
5.1.15	<i>Varios</i>	33
5.1.16	<i>Flujo de programas</i>	34
5.1.17	<i>Texto</i>	34
5.2	<i>Bibliografía</i>	35

1 Criterios y Expresiones frecuentes en Access

1.1 Introducción

Cuando se desea limitar los resultados de una consulta basada en los valores de un campo, se usan los criterios de consulta. Un criterio de consulta es una expresión que Access compara con los valores de los campos de consulta para determinar si se va a incluir o no el registro que contiene cada uno de los valores. Por ejemplo, = "Chicago" es una expresión que Access puede comparar con los valores de un campo de texto de una consulta. Si el valor para ese criterio en un registro determinado es "Chicago", Access incluye el registro en los resultados de la consulta.

Un criterio es similar a una fórmula. Es una cadena que puede estar formada por referencias de campo, [operadores](#) y [constantes](#). Los criterios de consulta también se conocen como expresiones.

En la tabla siguiente, se muestran algunos ejemplos de criterios y se explica su funcionamiento.

Criterio	Descripción
>25 y <50	Este criterio se aplica a campos Número, como Precio o UnidadesEnStock. El resultado incluye solamente aquellos registros en los que el campo Precio o UnidadesEnStock contiene un valor mayor que 25 y menor que 50 .
DiffFecha("aaaa", [FechaNacimiento], Fecha()) > 30	Este criterio se aplica a campos Fecha/Hora, como FechaNacimiento. En el resultado de la consulta, sólo se incluyen los registros en los que el número de años entre la fecha de nacimiento de una persona y la fecha actual es mayor que 30 .
Es Nulo	Este criterio se puede aplicar a cualquier tipo de campo para que se muestren los registros en los que el valor de campo sea nulo .

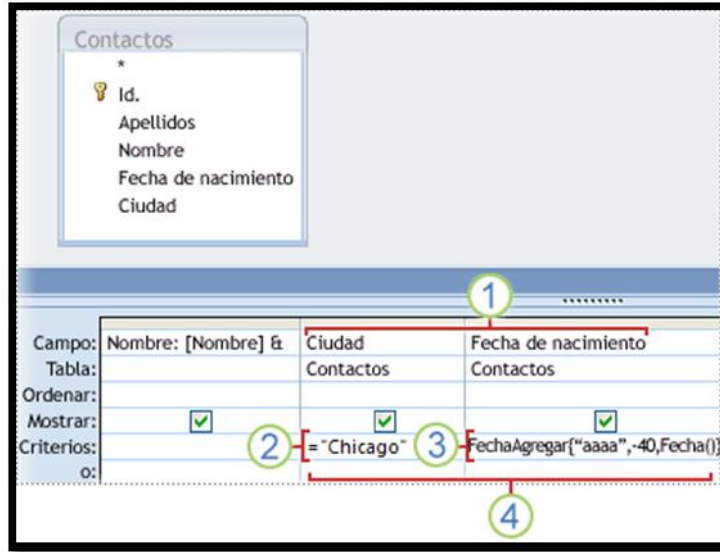
Los criterios pueden ser muy distintos entre sí. Algunos criterios son sencillos y usan operadores básicos (como "<") y constantes (como "50"). Otros son complejos y usan funciones ("como DiffFecha") o referencias de campo (como "[FechaNacimiento]"). Este tema proporciona varios ejemplos de criterios usados con frecuencia, agrupados por el tipo de datos al que se aplican. Puede usar los ejemplos proporcionados en este tema como punto de partida para especificar sus propios criterios.

1.2 Diseñando los criterios: la fila de criterios

1.2.1 Combinando criterios (Operador Y)

Los criterios que especifique para los distintos campos de la fila **Criterios** se combinan mediante el operador Y. Por ejemplo, podría especificar el criterio = "Chicago" para el campo Ciudad y el criterio < **AgregFecha("aaaa", -40, Fecha())** para el campo FechaNacimiento. Los dos criterios se interpretan juntos de la siguiente manera:

Ciudad = "Chicago" Y FechaNacimiento < AgregFecha("aaaa", -40, Fecha())



1. Los campos Ciudad y FechaNacimiento incluyen criterios.
2. Sólo los registros cuyo valor para el campo Ciudad sea Chicago se ajustarán al criterio.
3. Sólo los registros que tengan como mínimo 40 años de antigüedad se ajustarán al criterio.
4. Sólo los registros que cumplan los dos criterios se incluirán en los resultados.

1.2.2 Criterios alternativos (Operador O)

Si tenemos criterios alternativos, o dos conjuntos de criterios independientes donde basta con que se cumpla un conjunto, debemos usar las filas Criterios y o en la cuadrícula de diseño.



1. El criterio Ciudad se especifica en la fila Criterios.
2. El criterio FechaNacimiento se especifica en la fila o.

Los criterios especificados en las filas **Criterios** y **o** se combinan utilizando el operador O, como se muestra a continuación:

Ciudad = "Chicago" O FechaNacimiento < AgregFecha("aaaa", -40, Fecha())

Si necesitamos especificar más alternativas, usamos las filas situadas debajo de la fila **o**.

Hemos de tener en cuenta lo siguiente:

- Si los criterios son temporales o cambian a menudo, podemos filtrar el resultado de la consulta en lugar de estar modificando continuamente los criterios. Un filtro cambia el resultado de la consulta sin modificar el diseño de ésta.
- Si los campos de los criterios no cambian pero los valores que desea sí cambian con frecuencia, podemos crear una consulta de parámetros. Una consulta de parámetros solicita al usuario una entrada de datos y luego usa estos datos para crear los criterios de consulta.

1.3 Principales tipos de criterios y expresiones empleadas

1.3.1 Expresiones para campos Texto, Memo e Hipervínculo. Ejemplos

Los ejemplos siguientes son para el campo PaísRegión de una consulta basada en una tabla donde se almacenan datos de contacto. El criterio se especifica en la fila **Criterios** del campo en la cuadrícula de diseño como hemos visto.

Presentamos a continuación los siguientes ejemplos relacionados

Para incluir registros que...	Use este criterio	Resultado de la consulta
Coincidan exactamente con un valor, como China	"China"	Devuelve los registros en los que el campo PaísRegión tiene el valor China.
No coincidan con un valor, como México	No "México"	Devuelve los registros en los que el campo PaísRegión tiene un valor distinto de México.
Empiecen por la cadena especificada, como U	Como U*	Devuelve los registros correspondientes a todos los países o regiones cuyos nombres empiezan por "U", como Uruguay, Uganda, etc.
No empiecen por la cadena especificada, como U	No como U*	Devuelve los registros correspondientes a todos los países o regiones cuyos nombres empiezan por un carácter distinto de "U".
Contengan la cadena especificada, como Corea	Como "*Corea*"	Devuelve los registros correspondientes a todos los países o regiones que contienen la cadena "Corea".
No contengan la cadena especificada, como Corea	No como "*Corea*"	Devuelve los registros correspondientes a todos los países o regiones que no contienen la cadena "Corea".
Acaben con la cadena especificada, como "ina"	Como "*ina"	Devuelve los registros correspondientes a todos los países o regiones cuyos nombres acaban con "ina" como China y Argentina.
No acaben con la cadena especificada, como "ina"	No como "*ina"	Devuelve los registros correspondientes a todos los países o regiones cuyos nombres no acaban con "ina", como China y Argentina.
Contengan valores nulos (o que falten)	Es Nulo	Devuelve los registros en los que no hay ningún valor en el campo.
No contengan valores nulos	No es Nulo	Devuelve los registros en los que no falta el valor del campo.
Contengan cadenas de longitud cero	"" (un par de comillas)	Devuelve los registros en los que el campo tiene un valor en blanco (pero no nulo). Por ejemplo, los registros de ventas

		realizada a otro departamento podrían contener un valor en blanco en el campo PaísRegión.
No contengan cadenas de longitud cero	No ""	Devuelve los registros en los que el campo PaísRegión no tiene un valor en blanco.
Contengan valores nulos o cadenas de longitud cero.	"" O Es Nulo	Devuelve los registros donde no hay ningún valor en el campo, o bien, el campo tiene un valor en blanco.
Ni vacío ni en blanco	No es Nulo Y Sin ""	Devuelve los registros en los que el campo PaísRegión tiene un valor no en blanco y no nulo.
Vayan a continuación de un valor, como México, si se ordenan alfabéticamente	>= "México"	Devuelve los registros de todos los países o regiones a partir de México y posteriores por orden alfabético.
Estén incluidos en un intervalo específico, como de la A hasta la D	Como "[A-D]*"	Devuelve los registros correspondientes a los países o regiones cuyos nombres empiezan por las letras "A" a la "D".
Coincidan con uno de dos valores, tales como Estados Unidos o Reino Unido	"Estados Unidos" O "Reino Unido"	Devuelve los registros correspondientes a Estados Unidos y Reino Unido.
Contengan uno de los valores de una lista	En("Francia", "China", "Alemania", "Japón")	Devuelve los registros correspondientes a todos los países o regiones especificados en la lista.
Contengan ciertos caracteres en una posición específica en el valor del campo	Der([PaísRegión], 1) = "y"	Devuelve los registros correspondientes a todos los países o regiones cuyo nombre acaba con la letra "y".
Satisfagan requisitos de longitud	Longitud([PaísRegión]) > 10	Devuelve los registros correspondientes a los países o regiones cuyo nombre tiene más de 10 caracteres.
Coincidan con un modelo específico	Como "Chi??"	Devuelve los registros correspondientes a los países o regiones cuyos nombres tienen cinco caracteres de longitud y los tres primeros caracteres son "Chi", como China y Chile. Nota Los caracteres ? y _ cuando se usan en una expresión, representan un solo carácter (también se les denomina "caracteres comodín"). El carácter _ no se puede usar en la misma expresión junto con el carácter ? ni con el carácter comodín *. El carácter comodín _ se puede usar en una expresión que también contiene el carácter comodín %.

Breve descripción de las principales funciones texto, en función de su frecuencia de uso o utilidad.

- **Izq**: devuelve el principio de una cadena hasta una posición especificada
- **Der**: devuelve el final de una cadena hasta una posición especificada
- **Medio**: devuelve un fragmento de una cadena especificado por una posición inicial y una longitud
- **Longitud**: devuelve la longitud de una cadena
- **EnCad**: devuelve la primera posición de una cadena dentro de otra.
- **InStrRev**: igual que EnCad pero empezando desde el final
- **Minús**: devuelve una cadena en minúsculas
- **Mayús**: devuelve una cadena convertida a mayúsculas
- **Reemplazar**: reemplaza una subcadena por otra dentro de una cadena
- **RecortarIzq**: recorta los espacios iniciales de una cadena
- **RecortarDer**: recorta los espacios finales de una cadena
- **Recortar**: recorta los espacios iniciales y finales
- **StrReverse**: devuelve la cadena al revés
- **Cadena**: devuelve una cadena de caracteres repetidos tantas veces como se especifique
- **Espacio**: devuelve un número especificado de espacios
- **CompCadena**: sirve para comparar cadenas
- **Formato**: devuelve una expresión formateada conforme a unos parámetros dados, se utiliza sobre todo con fechas pero yo prefiero usar las expresiones de fecha que son más intuitivas que
- **ConvCadena**: convierte una cadena en función de una codificación, se utiliza para conversiones entre idiomas y codificación de cadenas

1.3.2 Separar nombre y apellidos en una tabla de personas

<http://josenrique.es/2011/06/separar-nombre-y-apellidos-en-una-tabla-de-personas/>

Caso, contamos con una tabla de personal en la que el nombre de la persona está codificado en la forma “Apellidos, Nombre” para muchos informes e incluso para algunas consultas preferiríamos tener estos datos en dos campos: uno para el nombre y otro para los apellidos ¿cómo podemos conseguirlo?

La forma más sencilla y rápida es utilizar una consulta de Access, pero previamente tendremos que crear los campos “nombre” y “apellidos” desde el diseño de la tabla. Cuando ya los tengamos separados podremos borrar el campo inicial u optar por dejarlo.

Una vez creados los campos necesitamos hacer una **consulta de actualización de datos**. Aunque es recomendable hacer siempre una prueba primero con una consulta normal y luego hacer la de actualización, así resulta mucho más sencillo detectar y corregir errores.

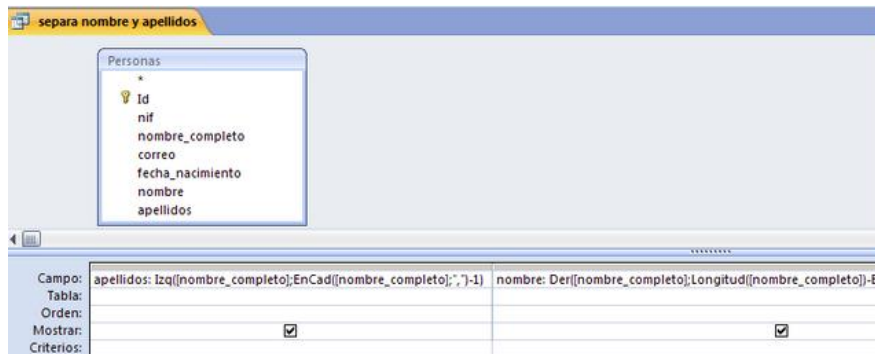
Empecemos con la consulta normal, creamos una consulta basada en la tabla **Personas** y sin seleccionar ningún campo tecleamos esto en la primera columna en el apartado **Campo**:

apellidos: Izq([nombre_completo];EnCad([nombre_completo];",-)-1)

Que “traducido” sería: muéstrame una columna y ponle de alias “Apellidos”, del campo “nombre_completo” muestra el contenido desde el comienzo, o sea empezando por la izquierda (Izq), hasta el lugar donde haya una coma, pero réstale un lugar para que la coma no salga. Si no sabes de donde salen estas funciones y quieres aprender un poco más sobre ellas, aquí tienes una [primera referencia](#).

En la segunda columna escribo esta otra expresión, no la explico para no alargar demasiado el artículo pero si tienes interés en aprender a utilizar las funciones te animo a intentar “traducirla” tu mismo.

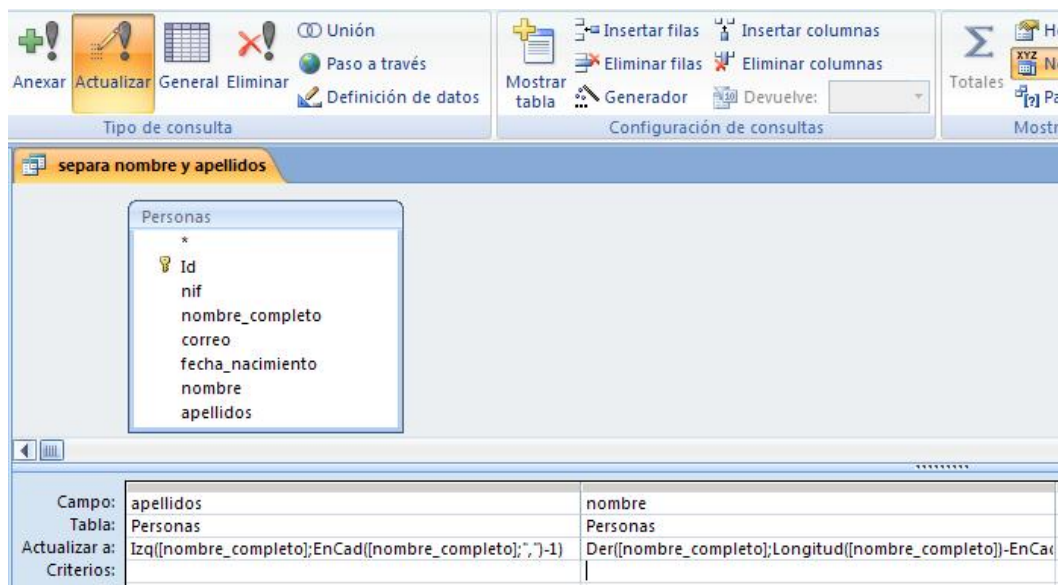
```
nombre:Der([nombre_completo];Longitud([nombre_completo])-
EnCad([nombre_completo];",")-1)
```



Ahora ejecutas la consulta y obtendrás una tabla de datos con dos columnas en las que el nombre y el apellido se encuentran separados.

apellidos	nombre
Ruiz Santos	Felipe
Péres Ramírez	José Manuel
Esteso Perdriza	Felipe
Peralias Estévez	Aniceto
*	

Ya hemos comprobado que nuestra consulta se ejecuta correctamente es el momento de pasar a la acción, para ello convertimos la consulta en una Consulta de Actualización de Datos. Lo siguiente es delicado así que hazlo con cuidado y te ahorrarás trabajo: borra los dos puntos que separan el alias de la función en la fila Campo, corta las funciones y la pegas en la fila Actualizar a, cada una en su columna correspondiente claro, pero dejando el alias donde estaba. En la fila Tabla debe aparecerte ahora el nombre de la tabla.



Por último ejecuta la consulta, acepta en el cuadro de dialogo que aparece y vete a ver la tabla **Personas** (no olvides guardar la consulta antes).

Personas						
Id	nif	nombre_completo	correo	fecha_nac	nombre	apellidos
4	28345676L	Ruiz Santos, Felipe	ruysantos@gmail.co	12/10/1967	Felipe	Ruiz Santos
5	32566453N	Péres Ramírez, José Manuel	perramirez@hotmail	21/03/1975	José Manuel	Péres Ramírez
6	27345567P	Esteso Pedriza, Felipe	esteso@hotmail.cor	01/06/1981	Felipe	Esteso Pedriza
7	28544636T	Peralias Estévez, Aniceto	peraliaseste@gmail	30/01/1963	Aniceto	Peralias Estévez

1.3.3 Expresiones "Valores de texto, parciales y coincidentes". Ejemplos

Expresión	Resultado
"Londres"	Para un campo CiudadDestinatario, pedidos enviados a Londres
"Londres" O "Hedge End"	Para un campo CiudadDestinatario, pedidos enviados a Londres o Hedge End
>="N"	Para un campo NombreCompañía, pedidos enviados a compañías cuyo nombre comienza por las letras de la N a la Z.
Como "S*"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por la letra S (ANSI-89)
Como "S%"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por la letra S (ANSI-92)
Der([IdPedido], 2)="99"	Para un campo IdPedido, pedidos con valores de Id que acaban en 99
Len([NombreCompañía]) > Val(30)	Para un campo NombreCompañía, pedidos para compañías cuyo nombre tiene más de 30 caracteres

1.3.4 Expresiones con "Parte de un valor de campo" (Como). Ejemplos

Expresión	Resultado
Como "S*"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por la letra S (ANSI-89)
Como "S%"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por la letra S (ANSI-92)
Como "*Importaciones"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre finaliza con la palabra "Importaciones" (ANSI-89)
Como "%Importaciones"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre finaliza con la palabra "Importaciones" (ANSI-92)
Como "[A-D]*"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por las letras de la A a la D (ANSI-89)
Como "[A-D]%"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre empieza por las letras de la A a la D (ANSI-92)
Como "*ar*"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre incluye la secuencia de letras "ar" (ANSI-89)
Como "%ar%"	Para un campo Destinatario, pedidos enviados a clientes cuyo nombre incluye la secuencia de letras "ar" (ANSI-92)
Como "Casa Dewe?"	Para un campo Destinatario, pedidos enviados al cliente con "Casa" como primera parte del nombre y un segundo nombre de 5 letras, de las cuales las cuatro primeras son "Dewe" y la última se desconoce (ANSI-89).

Como "Casa Dewe_"	Para un campo Destinatario, pedidos enviados al cliente con "Casa" como primera parte del nombre y un segundo nombre de 5 letras, de las cuales las cuatro primeras son "Dewe" y la última se desconoce (ANSI-92).
-------------------	--

1.3.5 Expresiones. Manipular valores de texto

Expresión	Resultado
NombreCompleto: [Nombre] & " " & [Apellidos]	Muestra el valor de los campos Nombre y Apellidos, separados por un espacio, en el campo NombreCompleto
Dirección2: [Ciudad] & " " & [Región] & " " & [CódigoPostal]	Muestra el valor de los campos Ciudad, Región y CódigoPostal, separados por espacios, en el campo Dirección2
ProductoInicial: Izq([NombreProducto], 1)	Muestra el primer carácter del valor del campo NombreProducto en el campo ProductoInicial
TipoCódigo: Der([CódigoVentaja], 2)	Muestra los dos últimos caracteres del valor del campo CódigoVentaja en el campo TipoCódigo
AreaCode: Medio([Teléfono], 2, 3)	Muestra los tres caracteres que comienzan por el segundo carácter del valor del campo Teléfono en el campo AreaCode

1.3.6 Expresiones para campo fecha/hora. Ejemplos

Los ejemplos siguientes son para el campo FechaPedido de una consulta basada en una tabla donde se almacena información de pedidos. El criterio se especifica en la fila **Criterios** del campo en la cuadrícula de diseño de la consulta.



Para incluir registros que...	Use este criterio	Resultado de la consulta
Coincidan exactamente con un valor, como 2/2/2006	#2/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar el 2 de febrero de 2006. No olvide rodear los valores de fechas con el carácter #, para que Access pueda distinguir entre fechas y cadenas de texto.
No coincidan con un valor, como 2/2/2006	No #2/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar en una fecha distinta al 2 de febrero de 2006.
Contengan valores anteriores a una fecha determinada, como el 2/2/2006	< #2/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar antes del 2 de febrero de 2006. Para ver las transacciones que tuvieron lugar en esa fecha o antes, use el operador <= en lugar de <.
Contengan valores posteriores a una fecha determinada, como el 2/2/2006	> #2/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar después del 2 de febrero de 2006.

		Para ver las transacciones que tuvieron lugar en esa fecha o después, use el operador >= en lugar de >.
Contengan valores que estén dentro de un intervalo de fechas	>#2/2/2006# Y <#4/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar entre el 2 de febrero de 2006 y el 4 de febrero de 2006. También puede usar el operador Entre para filtrar según un intervalo de valores, incluidos los extremos. Por ejemplo, Entre #2/2/2006# Y #4/2/2006# es lo mismo que >#2/2/2006# Y <#4/2/2006#.
Contengan valores que estén fuera de un intervalo	<#2/2/2006# O >#4/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar antes del 2 de febrero de 2006 o después del 4 de febrero de 2006.
Contengan uno de dos valores, como 2/2/2006 o 3/2/2006	#2/2/2006# O #3/2/2006#	Devuelve los registros de las transacciones que tuvieron lugar el 2 de febrero de 2006 o el 3 de febrero de 2006.
Contengan uno entre varios valores	En (#1/2/2006#, #1/3/2006#, #1/4/2006#)	Devuelve los registros de las transacciones que tuvieron lugar el 1 de febrero de 2006, el 1 de marzo de 2006 o el 1 de abril de 2006.
Contengan una fecha que sea de un mes específico (independientemente del año), como diciembre	ParcFecha("m", [FechaVenta]) = 12	Devuelve los registros de las transacciones que tuvieron lugar en diciembre de cualquier año.
Contengan una fecha que sea de un trimestre específico (independientemente del año), como el primer trimestre	ParcFecha("t", [FechaVenta]) = 1	Devuelve los registros de las transacciones que tuvieron lugar en el primer trimestre de cualquier año.
Contengan la fecha en curso	Fecha()	Devuelve los registros de las transacciones que tuvieron lugar en la fecha en curso. Si hoy es el 2/2/2006, verá los registros en los que el campo FechaPedido tiene el valor 2 de febrero de 2006.
Contengan la fecha de ayer	Fecha()-1	Devuelve los registros de las transacciones que tuvieron lugar en el día anterior a la fecha en curso. Si hoy es el 2/2/2006, verá los registros correspondientes al 1 de febrero de 2006.
Contengan la fecha de mañana	Fecha() + 1	Devuelve los registros de las transacciones que tendrán lugar en el día posterior a la fecha en curso. Si hoy es el 2/2/2006, verá los registros correspondientes al 3 de febrero de 2006.
Contengan fechas que sean de la semana en curso	ParcFecha("ss", [FechaVenta]) = ParcFecha("ss", Fecha()) Y Año([FechaVenta]) = Año(Fecha())	Devuelve los registros de las transacciones que tuvieron lugar durante la semana en curso. Una semana empieza en domingo y termina en sábado.
Contengan fechas que sean de la semana anterior	Año([FechaVenta])* 53 + ParcFecha("ss", [FechaVenta]) = Año(Fecha())* 53 +	Devuelve los registros de las transacciones que tuvieron lugar durante la semana pasada. Una semana empieza en domingo y termina en sábado.

	<code>ParcFecha("ss", Fecha()) - 1</code>	
Contengan fechas que sean de la semana próxima	<code>Año([FechaVenta])*53+ParcFecha("ss", [FechaVenta]) = Año(Fecha()*53+ParcFecha("ss", Fecha()) + 1</code>	Devuelve los registros de las transacciones que tendrán lugar la semana próxima. Una semana empieza en domingo y termina en sábado.
Contengan fechas que sean de los 7 últimos días	<code>Entre Fecha() Y Fecha()-6</code>	Devuelve los registros de las transacciones que tuvieron lugar durante los 7 últimos días. Si hoy es el 2/2/2006, verá los registros correspondientes al período del 24 de enero de 2006 al 2 de febrero de 2006.
Contengan una fecha que sea del mes en curso	<code>Año([FechaVenta]) = Año(Ahora()) Y Mes([FechaVenta]) = Mes(Ahora())</code>	Devuelve los registros correspondientes al mes en curso. Si hoy es el 2/2/2006, verá los registros correspondientes a febrero de 2006.
Contengan una fecha que sea del mes anterior	<code>Año([FechaVenta])*12 + ParcFecha("m", [FechaVenta]) = Año(Fecha()*12 + ParcFecha("m", Fecha()) - 1</code>	Devuelve los registros correspondientes al mes pasado. Si hoy es el 2/2/2006, verá los registros correspondientes a enero de 2006.
Contengan una fecha que sea del mes próximo	<code>Año([FechaVenta])*12 + ParcFecha("m", [FechaVenta]) = Año(Fecha()*12 + ParcFecha("m", Fecha()) + 1</code>	Devuelve los registros correspondientes al mes próximo. Si hoy es el 2/2/2006, verá los registros correspondientes a marzo de 2006.
Contengan fechas que sean de los 30 ó 31 últimos días	<code>Entre Fecha() Y AgregFecha("M", -1, Fecha())</code>	Los registros de ventas de un período de un mes. Si hoy es el 2/2/2006, verá los registros correspondientes al período del 2 de enero de 2006 al 2 de febrero de 2006.
Contengan una fecha que sea del trimestre en curso	<code>Año([FechaVenta]) = Año(Ahora()) Y ParcFecha("t", Fecha()) = ParcFecha("t", Ahora())</code>	Devuelve los registros correspondientes al trimestre en curso. Si hoy es el 2/2/2006, verá los registros correspondientes al primer trimestre de 2006.
Contengan una fecha que sea del trimestre anterior	<code>Año([FechaVenta])*4+ParcFecha("t",[FechaVenta]) = Año(Fecha()*4+ParcFecha("t",Fecha())-1</code>	Devuelve los registros correspondientes al trimestre anterior. Si hoy es el 2/2/2006, verá los registros correspondientes al último trimestre de 2005.
Contengan una fecha que sea del próximo trimestre	<code>Año([FechaVenta])*4+ParcFecha("t",[FechaVenta]) = Año(Fecha()*4+ParcFecha("t",Fecha())+1</code>	Devuelve los registros correspondientes al próximo trimestre. Si hoy es el 2/2/2006, verá los registros correspondientes al segundo trimestre de 2006.
Contengan fechas que sean del año en curso	<code>Año([FechaVenta]) = Año(Fecha())</code>	Devuelve los registros correspondientes al año en curso. Si hoy es el 2/2/2006, verá los registros correspondientes al año 2006.
Contengan una fecha que sea del año anterior	<code>Año([FechaVenta]) = Año(Fecha()) - 1</code>	Devuelve los registros de las transacciones que tuvieron lugar durante el año pasado. Si hoy es el 2/2/2006, verá los registros correspondientes al año 2005.

Contengan una fecha que sea del año próximo	$\text{Año}([\text{FechaVenta}]) = \text{Año}(\text{Fecha}()) + 1$	Devuelve los registros de las transacciones con fecha en el año próximo. Si hoy es el 2/2/2006, verá los registros correspondientes al año 2007.
Contengan una fecha que esté entre el 1 de enero y hoy (registros del año hasta la fecha)	$\text{Año}([\text{FechaVenta}]) = \text{Año}(\text{Fecha}()) \text{ Y } \text{Mes}([\text{FechaVenta}]) <= \text{Mes}(\text{Fecha}()) \text{ Y } \text{Día}([\text{FechaVenta}]) <= \text{Día}(\text{Fecha}())$	Devuelve los registros de las transacciones con fechas entre el 1 de enero del año en curso y la fecha actual. Si hoy es el 2/2/2006, verá los registros correspondientes al período del 1 de enero de 2006 al 2 de febrero de 2006.
Contengan una fecha ya pasada	$< \text{Fecha}()$	Devuelve los registros de las transacciones que tuvieron lugar antes de hoy.
Contengan una fecha del futuro	$> \text{Fecha}()$	Devuelve los registros de las transacciones que tendrán lugar después de hoy.
Filtren valores nulos (o que falten)	Es Nulo	Devuelve los registros en los que falta la fecha de la transacción.
Filtren valores no nulos	No es Nulo	Devuelve los registros en los que se conoce la fecha de la transacción.

1.3.7 Expresiones con Rangos de Valores (>, <, >=, <=, <>, o Entre...Y). Ejemplos

Expresión	Resultado
> 234	Para un campo Cantidad, números mayores de 234
< 1200.45	Para un campo PrecioUnidad, números menores de 1200,45
$>= \text{"Díaz"}$	Para un campo Apellidos, todos los nombres desde Díaz hasta el final del alfabeto
Entre #02.02.99# Y #01.12.99#	Para un campo FechaPedido, fechas del 2-Feb-99 al 1-Dic-99 (ANSI-89)
Entre '2/2/1999' Y '12/1/1999'	Para un campo FechaPedido, fechas del 2-Feb-99 al 1-Dic-99 (ANSI-92)

1.3.8 Expresiones con Valores que no coinciden (Negado). Ejemplos

Expresión	Resultado
Negado "EE.UU."	Para un campo PaísDestinatario, pedidos enviados a países que no sean EE.UU.
Negado 2	Para un campo Id, el empleado cuyo identificador no es 2
Negado T*	Para un campo Apellidos, los empleados cuyos nombres no empiezan por la letra "T" (ANSI-89)
Negado T%	Para un campo Apellidos, los empleados cuyos nombres no empiezan por la letra "T" (ANSI-92)

1.3.9 Expresiones con Valores de una lista (En). Ejemplos

Expresión	Resultado
En("Canadá", "Reino Unido")	Para un campo PaísDestinatario, pedidos enviados a Canadá o al Reino Unido
En(Francia, Alemania, Japón)	Para un campo País, empleados que viven en Francia, Alemania o Japón

1.3.10 Expresiones condicionales SiInm. Ejemplos

La siguiente tabla enumera ejemplos de expresiones que puede utilizar en los controles calculados.

Expresión	Descripción
=SiInm([Confirmado] = "Sí", "Pedido confirmado", "Pedido no confirmado")	Muestra el mensaje "Pedido confirmado" si el valor del campo Confirmado es Sí; en caso contrario, muestra el mensaje "Pedido no confirmado".
=SiInm(EsNulo([País]), " ", [País])	Muestra una cadena en blanco si el valor del campo País es Nulo (Null: valor que puede especificarse en un campo o utilizarse en expresiones o consultas para indicar datos desconocidos o ausentes. En Visual Basic, la palabra clave Null indica un valor Null. Algunos campos, como los de clave principal, no pueden contener Null.); en caso contrario, muestra el valor del campo País.
=SiInm(EsNulo([Región]), [Ciudad] & " " & [CódPostal], [Ciudad] & " " & [Región] & " " & [CódPostal])	Muestra los valores de los campos Ciudad y CódigoPostal si Región es Nulo; en caso contrario, muestra los valores de los campos Ciudad, Región y CódigoPostal.
=SiInm(EsNulo([FechaRequerida] - [FechaEnviado]), "Compruebe una fecha perdida ", [FechaRequerida] - [FechaEnviado])	Muestra el mensaje "Busque una fecha que falta" si el resultado de restar FechaEnviado de FechaRequerida es Nulo; en caso contrario, muestra la diferencia entre los valores de los campos FechaRequerida y FechaEnviado.

1.3.11 Un valor de campo vacío (Nulo o cadena de longitud cero). Ejemplos

Expresión	Resultado
Es Nulo	Para un campo RegiónDestinatario, pedidos de los clientes cuya RegiónDestinatario sea Nulo (Null: valor que puede especificarse en un campo o utilizarse en expresiones o consultas para indicar datos desconocidos o ausentes. En Visual Basic, la palabra clave Null indica un valor Null. Algunos campos, como los de clave principal, no pueden contener Null.) (vacío)
No es Nulo	Para un campo RegiónDestinatario, pedidos de los clientes cuya RegiónDestinatario contenga un valor
Notas	Para un campo Fax , pedidos de los clientes que no tienen una máquina de fax, lo que se indica mediante un valor de cadena de longitud cero (cadena de longitud cero: cadena que no contiene caracteres. Puede utilizar una cadena de longitud cero para indicar que sabe que no hay ningún valor para un campo. Para especificar una cadena de longitud cero, escriba dos comillas dobles sin espacio entre ellas (" ").) en el campo Fax en lugar de un valor Nulo (vacío)

1.3.12 Expresiones para otros campos

Tipo de campo	Cómo aplicar los criterios
Sí/No	En la fila Criterios, escriba Sí para incluir los registros que tengan activada la casilla de verificación. Escriba No para incluir los registros que no tengan activada la casilla de verificación.
Datos adjuntos	En la fila Criterios, escriba Es Nulo para incluir los registros que no contengan datos adjuntos. Escriba No es Nulo para incluir los registros que sí contengan datos adjuntos.
Búsqueda	<p>Hay dos tipos básicos de campos de búsqueda: los que buscan valores en un origen de datos existente (mediante una clave externa) y los que se basan en una lista de valores especificados al crearse el campo de búsqueda.</p> <p>Los campos de búsqueda que se basan en una lista de valores especificados son del tipo de datos Texto, y los criterios válidos son los mismos que para otros campos de texto.</p> <p>Los criterios que se pueden usar en un campo de búsqueda basado en valores de un origen de datos existente dependen del tipo de datos de la clave externa, y no del tipo de los datos que se buscan. Por ejemplo, suponga que tiene un campo de búsqueda que muestra los nombres de empleados, pero usa una clave externa de tipo de datos Número. Puesto que el campo almacena un número en lugar de texto, usted utilizará criterios válidos para números, es decir, >2.</p> <p>Si no conoce el tipo de datos de la clave externa, puede examinar la tabla de origen en la vista Diseño para determinar los tipos de datos de los campos. Para ello:</p> <ol style="list-style-type: none"> 1. Busque la tabla de origen en el panel de navegación. 2. Abra la tabla en la vista Diseño. Para ello: <ul style="list-style-type: none"> • Haga clic en la tabla y, a continuación, presione CTRL+ENTRAR. • Haga clic con el botón secundario del mouse en la tabla y, a continuación, haga clic en Vista Diseño. 3. El tipo de datos de cada campo se muestra en la columna Tipo de datos de la cuadrícula de diseño de la tabla.

2 Especial sobre el uso del carácter comodín

2.1 Introducción

Access admite dos juegos de caracteres comodín diferentes porque admite dos estándares para el lenguaje de consulta estructurado: ANSI-89 y ANSI-92. Generalmente, se utilizan los caracteres comodín ANSI-89 cuando se ejecutan consultas y operaciones de buscar y reemplazar en bases de datos de Access: archivos .mdb y .accdb. Los caracteres comodín ANSI-92 se utilizan cuando se ejecutan consultas en proyectos de Access (archivos de Access conectados a bases de datos de Microsoft SQL Server). Los proyectos de Access utilizan el estándar ANSI-92 porque SQL Server usa dicho estándar. En la tabla siguiente se ofrece un listado de los métodos o herramientas que puede utilizar para buscar y reemplazar datos, y se indica el estándar ANSI utilizado con cada herramienta.

Método o herramienta de búsqueda	Tipo de archivo en el que se realiza la búsqueda	Juego de caracteres comodín utilizado
Cuadro de diálogo Buscar y reemplazar	Base de datos de Access (.archivos .mdb y .accdb)	ANSI-89
Cuadro de diálogo Buscar y reemplazar	Proyecto de Access (archivos .adp y .accdp)	ANSI-92
Consulta de selección o actualización	Base de datos de Access (archivos .mdb y .accdb)	ANSI-89
Consulta de selección o actualización	Proyecto de Access (archivo .adp y .accdp)	ANSI-92
Cuadro de diálogo Buscar y reemplazar , consulta de selección o actualización	Base de datos de Access configurada para su uso con el estándar ANSI-92	ANSI-92

2.2 Uso de carácter comodín en cuadro de diálogos **Buscar y Reemplazar** y **Consultas de selección y actualización (ANSI-89)**.

Debemos utilizar este juego de caracteres comodín cuando use el cuadro de diálogo **Buscar y reemplazar** para buscar y reemplazar datos en una base de datos o en un proyecto de Access. *Estos caracteres se utilizan también cuando se ejecutan consultas de selección y actualización, pero no se utilizan en las consultas ejecutadas en un proyecto de Access¹.*

Carácter	Descripción	Ejemplo
*	Hace coincidir cualquier número de caracteres. Puede utilizar el asterisco (*) en cualquier sitio de una cadena de caracteres.	qu* encuentra qué, quién y quizás pero no aquellos ni aunque.
?	Hace coincidir cualquier carácter alfabético individual.	B?l encuentra bala, billete y bola.
[]	Hace coincidir cualquier carácter individual incluido entre los corchetes.	B[ao]l encuentra bala y bola pero no billete.
!	Hace coincidir cualquier carácter que no se encuentre entre los corchetes.	r[!oc]a encuentra risa y rema pero no roca ni rosa.
-	Hace coincidir cualquier carácter incluido en un intervalo de caracteres. Debe especificar el intervalo en orden ascendente (de A a Z, no de Z a A).	b[a-c]d encuentra bad, bbd y bcd.
#	Hace coincidir cualquier carácter numérico individual.	1#3 encuentra 103, 113 y 123.

¹ Consulta de selección o actualización asociadas o ejecutadas a un proyecto de Access (archivo .adp y .accdp)

2.3 Uso de carácter comodín en cuadro de diálogos **Buscar y Reemplazar** y consultas de selección y actualización en proyectos Access (ANSI-92).

Es el caso menos frecuente para el usuario general y se usa cuando ejecutamos consultas de selección y actualización en proyectos de Access (archivos .adp)

Carácter	Descripción	Ejemplo
%	Hace coincidir cualquier número de caracteres. Puede utilizarse como el primero o el último carácter de la cadena de caracteres.	qu% encuentra qué, quién y quizás pero no aquellos ni aunque.
-	Hace coincidir cualquier carácter alfabético individual.	R_sa encuentra rosa, risa y rusa.
[]	Hace coincidir cualquier carácter individual incluido entre los corchetes.	B[ao]l encuentra bala y bola pero no billete.
^	Hace coincidir cualquier carácter que no se encuentre entre los corchetes.	r[^oc]a encuentra risa y rema pero no roca ni rosa.
-	Hace coincidir cualquier carácter incluido en un intervalo de caracteres. Debe especificar el intervalo en orden ascendente (de A a Z, no de Z a A).	b[a-c]d encuentra bad, bbd y bcd.

2.4 Tipos de datos que se pueden buscar mediante caracteres comodín

Cuando se diseña una tabla, se define un tipo de datos para cada campo de la tabla. Por ejemplo, se define el tipo de datos Fecha/Hora para los campos que contienen información de fecha. En esta tabla se ofrece un listado de los tipos de datos que puede buscar mediante caracteres comodín. Recuerde que, en algunos casos, puede utilizar caracteres comodín en el cuadro de diálogo **Buscar y reemplazar** pero no en las consultas, y viceversa.

Tipo de datos	Usar en ...
Texto	Cuadro de diálogo Buscar y reemplazar , consultas
Memo	Cuadro de diálogo Buscar y reemplazar , consultas
Número	Cuadro de diálogo Buscar y reemplazar , consultas
Fecha/Hora	Cuadro de diálogo Buscar y reemplazar , consultas Nota La configuración regional puede afectar al modo en que se utilizan los caracteres comodines.
Moneda	Cuadro de diálogo Buscar y reemplazar , consultas
Autonumérico	Cuadro de diálogo Buscar y reemplazar , consultas
Objeto OLE	Ninguno.
Sí/No	Consultas, pero no es necesario. Vea las notas al final de esta sección para obtener más información.
Hipervínculo	Cuadro de diálogo Buscar y reemplazar , consultas
Asistente para búsquedas	Según el tipo de datos del campo de origen.

3 Especial: Operando con campo fecha - hora

3.1 Introducción, ¿Para qué las funciones fecha y hora?

Problemas y cuestiones relacionadas con las funciones fecha

- ¿Cómo puedo determinar una fecha que es 50 días laborables después de otra fecha? ¿Qué pasa si quiero excluir los días festivos?
- ¿Cómo puedo determinar el número de días laborables entre dos fechas?
- Tengo 10.000 fechas diferentes correspondientes a los tickets de ventas del presente ejercicio, ¿Cómo puedo escribir fórmula para extraer de cada fecha el mes, año, día del mes y día de la semana?
- Para cada contrato laboral de los trabajadores eventuales de la temporada verano-otoño tengo la fecha de alta y la de baja, ¿Cómo puedo determinar el número de meses en que cada trabajador ha estado contratado?
- ¿Cuál es la antigüedad de mi inventario de productos?
- ¿Cómo puedo determinar qué día es 25 días laborables después de la fecha actual (incluyendo festivos)?
- ¿Cómo puedo determinar qué día es 21 días laborables después de la fecha actual incluyendo festivos pero excluyendo la navidad y año nuevo?
- Determinar la edad exacto en años de nuestros empleados
- ¿Cuántos días (incluyendo festivos) hay entre el 10-07-2011 y 15-08-2012?
- ¿Cuántos días (incluyendo festivos pero excluyendo navidad y fin de años) hay entre el 10 de julio de 2011 y 15 de agosto de 2012?

Problemas y cuestiones relacionadas con las funciones horas

- Estimar el tiempo dedicado a cada actividad según el registro de partes de trabajo rellena por cada operario de fábrica.
- Calcular los tiempos de reparto que ha tenido cada camión diariamente según el análisis de ruta que arroja nuestro GPS.
- Como controlar y gestionar la información contenida en un reloj de registro de personal con entradas y salidas.
- Nuestra TPV graba los tickets de nuestro PUB en el cual se registra no solo el importe sino la fecha y hora de cada consumición. Queremos analizar esta información para definir nuestra estrategia de marketing basada en Happy hour y por lo tanto es necesario extraer la información no solo sobre día de la semana (jueves, viernes, etc) y del mes (primera semana, segunda semana del mes, etc..) sino también las diferentes franjas horarias, para analizar los momentos de escasa actividad y consumo e incentivar estas franjas.
- Determinar el número de horas que ha trabajado un empleado.
- Sumar una cantidad de horas a un total de horas trabajadas.
- En general para resolver problemas relacionados con unidades de tiempo, para calcular horas de espera, tiempo trabajado, descansos, etc.

En cualquiera de los casos comentados necesitamos trabajar con las funciones fecha y hora.

3.2 Función `DifFecha("aaaa", [FechaNacimiento], Fecha()) > 30`

3.2.1 Objetivo de la función `DifFecha` o `DateDiff`

La función `DifFecha` nos permite calcular el intervalo de tiempo transcurrido entre dos fechas. Este intervalo de tiempo pueden ser días, meses, semanas, años, etc. Lo importante a la hora de utilizar la función `DifFecha` es indicar el intervalo adecuado.

3.2.2 Sintaxis de la función

`DifFecha ("intervalo";[fecha 1];[fecha 2];primer día semana; primera semana)`

La sintaxis de la función tiene 5 argumentos: los tres primeros son obligatorios y los dos últimos son opcionales.

Argumento	Descripción
<i>intervalo</i>	Requerido. Expresión de cadena que es el intervalo de tiempo que se utiliza para calcular la diferencia entre <i>fecha1</i> y <i>fecha2</i> .
<i>fecha1, fecha2</i>	Requerido. Variant (Date) . Las dos fechas que se desean utilizar en el cálculo.
<i>primerdíasemana</i>	Opcional. Constante que especifica el primer día de la semana. Si no se especifica, se supone que es el domingo.
<i>primerasemanaaño</i>	Opcional. Constante que especifica la primera semana del año. Si no se especifica, se supone que es la semana en la que transcurre el 1 de enero.

El intervalo es el primer argumento de la función, hemos de escribirlo entre comillas, pudiendo disponer de los siguientes intervalos:

- "d" días
- "w" día de la semana (por ejemplo miércoles transcurridos entre las dos fechas)
- "ww" semanas
- "m" meses
- "q" trimestres
- "y" días del año (transcurridos entre dos fechas, es lo mismo que especificar d)
- "yyyy" años
- "h" horas
- "n" minutos
- "s" segundos

ARGUMENTOS OPCIONALES:

Primer día de la semana, es una constante numérica, por defecto es 1, y corresponde al domingo, ya que se rige por el calendario anglosajón, cuyas semanas comienzan en domingo. Si queremos especificar que el primer día de la semana es el lunes escribiríamos un 2).

Primera semana, en este argumento se indica cual es la primera semana del año. Si no es especificamos nada, Access toma como primera semana del año la semana del 1 de enero.

3.3 Formato fecha y hora

3.3.1 Introducción. Fecha hora como numero

En Access, los valores de fecha y hora pueden aparecer en diversos formatos, como un formato europeo (28.11.2006 ó 28-11-2006), un formato surasiático (28/11/2006) o el formato de EE.UU. (11/28/2006).

Independientemente de cómo Access aplique formato a los valores de fecha y hora e independientemente de cómo se realice la entrada de los valores de fecha y hora, **Access almacena estos valores como números de punto flotante de precisión doble: se trata de un sistema que también se denomina fechas de serie.** En esta ilustración se representa un valor de fecha y hora de serie típico.



La parte correspondiente al entero del valor, situada a la izquierda del separador decimal, representa la fecha.

El número en la ilustración representa las 9 p.m. del 24 de diciembre de 2003. El componente de fecha es el número de días enteros que han transcurrido desde la fecha inicial o "base" del 30/12/1899. En este ejemplo, han transcurrido 37.979 días desde el 30/12/1899 hasta el 24/12/2003. El componente de hora es una fracción de un día de 24 horas. Por lo tanto, un valor de 0,875 multiplicado por 24 horas es igual a 21 horas o 9:00 p.m.

Los valores negativos en el componente de fecha representan fechas anteriores a la fecha base. Por ejemplo, el valor -1 como componente de fecha equivale a un día anterior a la fecha base, es decir, 29/12/1899.

El intervalo de valores de fecha válidos va de -657.434 (1 de enero de 100 D.C.) a 2.958.465 (31 de diciembre de 9999 D.C.). El intervalo de valores de hora válidos es de .0 a .9999, es decir, 23:59:59.

Al almacenar los valores de fecha y hora como números, se pueden realizar numerosos cálculos con las fechas y horas. Por ejemplo, se puede calcular el número total de horas trabajadas (ficha de presencia) o la antigüedad de una factura.

3.3.2 Formatos de fecha y hora personalizados. Día de la semana, del mes, semana del año, etc

Podremos emplear formatos personalizados para los campos de fecha y hora. Recogemos a continuación los más frecuentes

Carácter	Descripción
d o dd	Muestra el día del mes con uno o dos dígitos. Para un solo dígito, use un solo marcador de posición; para dos dígitos, use dos marcadores de posición.
ddd	Abrevia el día de la semana a tres letras. Por ejemplo, lunes aparecería como lun.
dddd	Muestra el nombre completo de todos los días de la semana.
ddddd	Muestra el formato de fecha corta.
dddddd	Muestra el formato de fecha larga.
s	Muestra el número del día de la semana. Por ejemplo, lunes aparecería como 2.
m o mm	Muestra el mes como un número de un dígito o de dos dígitos.
mmm	Abrevia el nombre del mes con tres letras. Por ejemplo, octubre aparecería como oct.
mmmm	Muestra el nombre completo de todos los meses.
t	Muestra el número del trimestre del calendario actual (1-4). Por ejemplo, para el mes de mayo, Access muestra el número 2 como valor de trimestre.
a	Muestra el día del año, del 1 al 366.
aa	Muestra los dos últimos dígitos del año.

	Nota Se recomienda escribir y mostrar los cuatro dígitos de los años.
aaaa	Muestra todos los dígitos de un año, de 0100 a 9999.
Separador de hora	Controles donde Access coloca el separador para las horas, los minutos y los segundos. Use el separador definido en la configuración regional de Windows. Para obtener información sobre esta configuración, vea la siguiente sección Cómo la configuración regional de Windows afecta a las fechas y horas.
h o hh	Muestra la hora con uno o dos dígitos.
n o nn	Muestra los minutos con uno o dos dígitos.
s o ss	Muestra los segundos con uno o dos dígitos.
hhhh	Muestra el formato de hora larga.

3.4 Ejemplos de manipulación y cálculo con campo fechas

3.4.1 Ejemplo

Expresión	Descripción
=Fecha()	Utiliza la función Fecha para mostrar la fecha actual en el formato <i>mm-dd-aaaa</i> , donde <i>mm</i> es el mes (del 1 al 12), <i>dd</i> es el día (del 1 al 31) y <i>aaaa</i> es el año (de 1980 a 2099).
=Formato(Ahora(), "ss")	Utiliza la función Formato (Format) para presentar el número de la semana del año que representa la fecha actual, donde <i>ss</i> es 1 a 53.
=ParcFecha("yyyy", [FechaPedido])	Utiliza la función ParcFecha (DatePart) para presentar los cuatro dígitos del año del valor del campo FechaPedido.
=SumFecha("a", -10, [FechaPrometida])	Utiliza la función SumFecha (DateDif) para presentar una fecha que sea 10 días antes del valor del campo FechaPrometida.
=DiffFecha("d", [FechaOrden], [FechaEnviado])	Utiliza la función DiffFecha para presentar la varianza en días entre los valores de los campos FechaOrden y FechaEnviado.

3.4.2 Ejemplo

Expresión	Resultado
#02.02.00#	Para un campo ShippedDate, pedidos enviados el 2 de febrero de 2000 (ANSI-89)
'02.02.00'	Para un campo ShippedDate, pedidos enviados el 2 de febrero de 2000 (ANSI-92)
Fecha()	Para un campo FechaRequerida, pedidos para hoy
Entre Fecha() Y AgregFecha("m", 3, Fecha())	Para un campo FechaRequerida, pedidos requeridos entre hoy y tres meses a partir de hoy
< Fecha() - 30	Para un campo FechaPedido, pedidos con una antigüedad de más de 30 días
Año([FechaPedido]) = 1999	Para un campo FechaPedido, pedidos con fechas de 1999
ParcFecha("t", [PedidoFecha]) = 4	Para un campo FechaPedido, pedidos para el cuarto trimestre del calendario

SerieFecha(Año ([FechaPedido]), Mes([FechaPedido]) + 1, 1) - 1	Para un campo FechaPedido, pedidos para el último día de cada mes
Año([FechaPedido])=Año(Ahora()) Y Mes([FechaPedido])=Mes(Ahora())	Para un campo FechaPedido, pedidos para el año y mes actuales

3.4.3 Ejemplo

Por ejemplo, tenemos una tabla de pedidos, y queremos saber cuántos días han transcurrido entre la fecha de pedido y la fecha de entrega. Creamos una consulta en la que utilizamos la función DiffFecha con la siguiente expresión:

Días transcurridos:DiffFecha("d";[Fecha_Pedido];[Fecha_Entrega])

Campo:	Fecha_Pedido	Fecha_Entrega	Días transcurridos: DiffFecha("d";[fecha_Pedido];[fecha_Entrega])
Tabla:	Pedidos	Pedidos	
Orden:			
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:			
o:			

Podríamos haber utilizado también el intervalo "y" para realizar la misma consulta. El resultado sería:

	Fecha_Pedido	Fecha_Entrega	Días transcurridos
	domingo, 27 de enero de 2008	domingo, 27 de abril de 2008	91
	jueves, 10 de enero de 2008	lunes, 21 de abril de 2008	102
	domingo, 20 de enero de 2008	lunes, 21 de abril de 2008	92
	jueves, 10 de enero de 2008	lunes, 21 de abril de 2008	102

Si nos interesa saber las semanas transcurridas entre ambas fechas indicaríamos como intervalo "ww", de esta forma contamos el número de domingos que hay entre la fecha de pedido y la fecha de entrega:

Semanas transcurridas:DiffFecha("ww";[Fecha_Pedido];[Fecha_Entrega])

Obtenemos:

	Fecha_Pedido	Fecha_Entrega	Semanas transcurridas
	domingo, 27 de enero de 2008	domingo, 27 de abril de 2008	13
	jueves, 10 de enero de 2008	lunes, 21 de abril de 2008	15
	domingo, 20 de enero de 2008	lunes, 21 de abril de 2008	13
	jueves, 10 de enero de 2008	lunes, 21 de abril de 2008	15

Si utilizamos el intervalo "w", día de la semana, y la primera fecha es jueves, contaremos el número de jueves transcurridos entre ambas fechas, sin tener en cuenta el primer jueves de la fecha 1. Es otra manera de contar las semanas transcurridas entre dos fechas.

3.4.4 Ejemplo

Expresión	Resultado
TiempoRetraso: DiffFecha("d", [FechaPedido], [FechaEnvío])	Muestra el número de días entre la fecha de pedido y la fecha de envío en el campo TiempoRetraso
YearHired: ParcFecha("aaaa", [HireDate])	Muestra el año en el que se contrató a cada empleado en el campo YearHired
MonthNo: ParcFecha("M", [FechaPedido])	Muestra el número del mes en el campo MonthNo
PriorDate: Fecha() - 30	Muestra la fecha 30 días anterior a la fecha actual en el campo PriorDate

4 Empleo de funciones de agregado de dominio frecuentes

4.1 Introducción

De forma sintética, las funciones de agregados son expresiones que realizan un recuento, suman y buscan valores de manera selectiva mediante funciones de agregado de dominio.

En concreto, las funciones agregadas de dominio permiten calcular estadísticas relativas sobre conjuntos de registros (un dominio). Por ejemplo, podemos usar una función agregada para contar el número de registros de un conjunto determinado de registros o para determinar el promedio de los valores de un campo determinado.

Así las funciones de agregado por tanto permiten obtener un resultado basado en los valores contenidos en una columna de una tabla, son funciones que sólo se pueden utilizar en una consulta de resumen ya que obtienen un 'resumen' de los valores contenidos en las filas de la tabla.

Es interesante destacar que el valor nulo no equivale al valor 0, las funciones de resumen no consideran los valores nulos mientras que consideran el valor 0 como un valor, por lo tanto en el promedio y la desviación estándar los resultados no serán los mismos con valores 0 que con valores nulo.

Destacar además que los valores a sumar, contar o determinar y el campo condición no necesariamente tiene que estar en el origen de datos en el que se basa la consulta, el formulario o informe. Por ejemplo, podemos mostrar el número de pedidos de la tabla Pedidos en un control calculado en un formulario basado en la tabla Productos.

Se usa un tipo de función denominado función de agregado de dominio para realizar un recuento o sumar valores de manera selectiva. Un "dominio" se compone de uno o varios campos en una o varias tablas, o bien, de uno o varios controles ubicados en uno o varios formularios o informes. Por ejemplo, se pueden hacer coincidir los valores de un campo de tabla con los valores de un control ubicado en un formulario

En resumen, se usa un tipo de función denominada función de agregado de dominio para realizar un recuento o sumar valores de manera selectiva. Un "dominio" se compone de uno o varios campos en una o varias tablas, o bien, de uno o varios controles ubicados en uno o varios formularios o informes. Por ejemplo, se pueden hacer coincidir los valores de un campo de tabla con los valores de un control ubicado en un formulario.

Prácticamente todas ellas son muy similares:

```
NombredelaFuncion("Campodelatabla";"Tabla";"Condicion")
```

Ejemplo disponemos de una tabla que vamos a llamar Clientes con los siguientes campos:

- IdCliente (Autonumérico)
- NombreCliente (Texto)
- DNICliente (Texto)
- FechaNacimiento (Fecha/Hora)
- Edad (Numero)

Para probarlo creamos un formulario independiente, con **un simple cuadro de texto**, que vamos a llamar **txtResultados**, y **un botón de comando**. Vamos a comenzar por intentar averiguar cuantos registros tenemos en nuestra tabla y para ello vamos a usar la función agregada de dominio DCount.

DCount("CampodelaTabla","Tabla","Condicion"), pero para este primer ejemplo no vamos a poner ninguna condición con lo que nos quedaría así:

```
DCount("IdCliente","Clientes").
```


Si hacemos click en el botón, nos aparecerá en nuestro cuadro de texto el número de registros de la tabla.

Como todas son muy parecidas, ahora vamos a ver DSum, vamos a sumar las edades de todos nuestros clientes, para ello en el botón de comando borramos lo que teníamos y ponemos:

Me.txtResultados = DSum("Edad","Clientes")

Vamos a obtener el promedio de edad de nuestros clientes, para ello dividiremos la suma de sus edades entre el número de clientes

Me.txtResultados = DSum("Edad","Clientes") / DCount("IdCliente","Clientes").

Funciona. Pero Access incluye una función para calcular el promedio, se trata de Avg. Probamos ahora poniendo

Me.txtResultados = DAvg("Edad","Clientes")

Y obtendremos el mismo resultado que antes.

Ahora vamos a descubrir que edad es la más alta de todas las de la tabla. Utilizaremos DMax

Me.txtResultados = DMax("Edad","Clientes")

La menor de todas con DMin

Me.txtResultados = DMin("Edad","Clientes").

Vamos a introducir condiciones, y a conocer **DLookup**. Esta función nos sirve para encontrar un campo en una tabla. El objetivo va a ser conocer el nombre del cliente que tengamos en la tabla cuyo IdCliente sea igual a 3.

La condición sería **"IdCliente = 3"**

Por tanto como lo que buscamos es el nombre del cliente quedaría así:

DLookup("NombreCliente","Clientes","IdCliente = 3")

Lo ponemos ahora en el botón de comando

Me.txtResultados = DLookup("NombreCliente","Clientes","IdCliente = 3")

Al hacer click, obtendremos el cliente que se corresponda con el numero 3.

4.2 Función DSuma ()

Se usa para calcular la suma de un conjunto de valores de un conjunto especificado de registros (un dominio). Se utiliza en un módulo de Visual Basic para Aplicaciones (VBA), en una macro, en una expresión de consulta o en un control calculado.

Sintaxis:

DSuma(*expr*; *dominio*; [*criterios*])

La sintaxis de la función DSuma tiene los siguientes argumentos:

Argumento	Descripción
<i>expr</i>	Requerido. Expresión que identifica el campo numérico cuyos valores desea calcular. Puede ser una expresión de cadena que identifique un campo de una tabla o de una consulta, o puede ser una expresión que realice un cálculo sobre los datos de ese campo. En <i>expr</i> , puede incluir el nombre de un campo de una tabla, un control de un formulario, una constante o una función. Si <i>expr</i> incluye una función, ésta puede estar integrada o puede ser definida por el usuario, pero no puede ser otra función de agregado de dominio ni una función de agregado de SQL.

dominio	Requerido. Expresión de cadena que identifica el conjunto de registros que constituye el dominio. Puede ser un nombre de tabla o un nombre de consulta de una consulta que no requiera un parámetro.
critérios	Opcional. Expresión de cadena utilizada para restringir el intervalo de los datos en los que se ejecuta la función DSuma . Por ejemplo, critérios con frecuencia es equivalente a la cláusula WHERE en una expresión SQL, sin la palabra WHERE. Si se omite critérios , la función DSuma evalúa expr para todo el dominio. Cualquier campo que esté incluido en critérios también debe ser un campo de dominio ; en caso contrario, la función DSuma devuelve un valor Null .

=**DSuma**("[Importe de pedido]", "[Pedidos]", "[Id. de cliente] = 'RATTC'") Usa la función **DSuma** para devolver el total de los valores del campo Importe de pedido de la tabla denominada Pedidos donde el Id. de cliente sea RATTC.

En el caso de que estemos trabajando con campos alfanuméricos, tenemos que tener en cuenta que el parámetro de condición debe estar entrecomillado.

Es decir, supongamos la función:

DSum("[cantidaddepreciacion"],"Depreciacion","[codID]=" & [codID])

donde cantidaddepreciacion es la variable que estamos sumando en un subformulario, de la tabla Depreciacion y la condición es que sume solamente los registros que tienen el mismo código codID, todo funciona perfecto cuando el codID es un auto numérico, pero si el codID fuera un alfanumérico (texto) tenemos que encerrar el último parámetro codid entre comilla simple, de esta manera:

DSum("[cantidaddepreciacion"],"Depreciacion","[codID]=" & [codID] & """)

4.3 Función DlookUp ()

DlookUp es una función predefinida de Access, como una fórmula a la que se le pasan ciertos datos y ella se encarga de hacer los cálculos necesarios y devolvernos el resultado ahorrándonos de esta manera un montón de pasos, es decir, en un contexto sencillo sería como decirle "Búscame el campo X en la tabla Z que coincida con el criterio Y".

Así supongamos que en una tabla de artículos disponemos de tres campos "Código, Descripción e Importe" y en alguna parte de nuestra base de datos queremos buscar el precio de un artículo concreto cuyo código es el 002568, para esto sirve.

DlookUp ("Importe", "T_Articulos", "Codigo= 002568"), la función es así de sencilla

¿Cómo y Donde se usa?

Como buena función de dominio que es, la podemos usar en varios apartados de nuestra base de datos como en un cuadro de texto calculado, una consulta, o con código VBA.

Ejemplo en un formulario

Tenemos un formulario en blanco, en este caso sin orígenes de datos con dos campos de texto, uno para poner el código y el otro para que lo calcule con nuestro DlookUp, aquí tenemos la vista diseño del mismo.

Codigo Artículo	Independiente
Importe Artículo	=DBÚsq("Importe";"T_Articulos";"Codigo= " & [txtCodigo] & "")

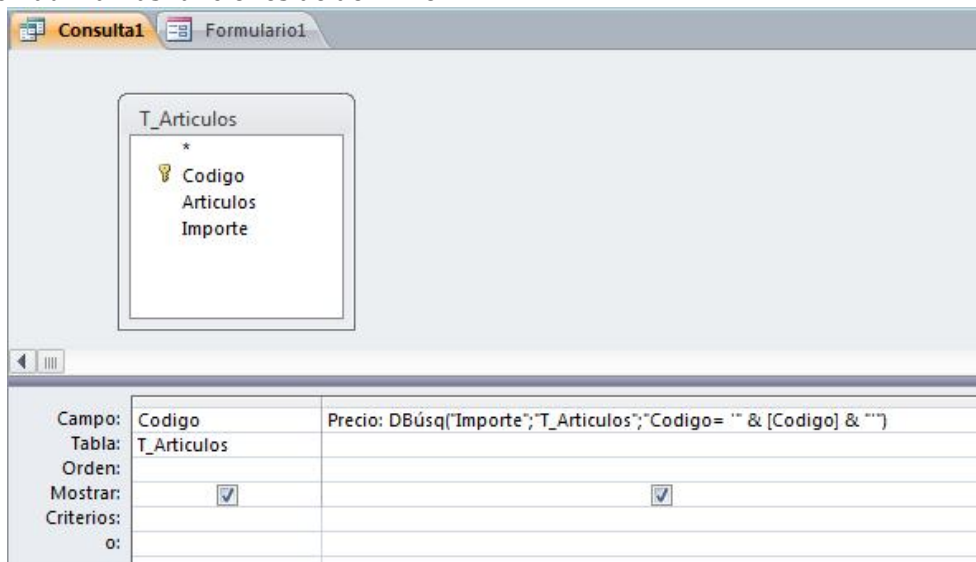
Como vemos el último dato “parámetro” del criterio lo que hace es leer lo que aparece en el primer cuadro de texto. Aunque el ejemplo es muy simple, si cambiamos el código del primer cuadro de texto no buscará automáticamente su importe, en este caso he añadido, a su evento “Después de actualizar”, la siguiente línea mediante VBA “Me.Recalc” para que vuelva a recalcular los cambios:

Codigo Artículo

Importe Artículo

Ejemplo en una consulta

Disponemos de una sola tabla sobre la cual hemos realizado la siguiente consulta con origen de datos la tabla “T_Articulos” y hemos añadido un campo calculado “Precio”, **en este ejemplo carece de sentido**, lo que importa es que veamos que en una consulta también se pueden utilizar las funciones de dominio:



Esto daría, el siguiente resultado:

Codigo	Precio
002568	0,85
003548	0,75
004998	0,5
006465	8
006468	3
006634	0,2
*	

4.4 Función DCont ()

La función DCont permite determinar el número de registros de un conjunto especificado de registros (un dominio). La función DCont se utiliza en un módulo de Visual Basic para Aplicaciones (VBA), en una macro, en una expresión de consulta o en un control calculado.

Por ejemplo, podríamos usar la función DCont en un módulo para devolver el número de registros de una tabla Pedidos que se corresponde a los pedidos realizados en una fecha determinada.

La función **DCont** no cuenta los registros que contengan valores **Null** en el campo que *expr* hace referencia a no ser que *expr* sea el carácter comodín asterisco (*). Si usa un asterisco, la función **DCont** calcula el número total de registros, incluyendo aquellos que contienen campos **Null**.

=**DCuenta**("[Retirados]", "[Activos]", "[Retirados]=Sí") Usa la función **DCuenta** para devolver el número de valores Sí en el campo Retirados (campo de tipo Sí/No) de la tabla denominada Activos

4.5 Función DBúsq ()

La función **DBúsq** permite obtener el valor de un campo en un conjunto especificado de registros (un dominio). Por tanto podemos mostrar el valor de un campo que no se encuentra en el origen de registros en que está basado el formulario o informe. Por ejemplo, supongamos que contamos con un formulario basado en una tabla Detalles de pedido. El formulario muestra los campos IdPedido, IdProducto, PrecioUnidad, Cantidad y Descuento. Sin embargo, el campo NombreProducto está en otra tabla, la tabla Productos. Puede utilizar la función **DBúsq** en un control calculado para mostrar el NombreProducto en el mismo formulario.

=**DBúsq**("[Nombre de contacto]", "[Proveedores]", "[Id. de proveedor] = " & Formularios("Proveedores")("[Id. de proveedor]")) Usa la función **DBúsq** para devolver el valor del campo Nombre de contacto de la tabla denominada Proveedores donde el valor del campo Id. de proveedor de la tabla coincida con el valor del control Id. de proveedor ubicado en el formulario denominado Proveedores.

4.6 Funciones DMín (DMin) y DMáx (DMax)

Podemos usar las funciones **DMín (DMin)** y **DMáx (DMax)** para calcular los valores mínimo y máximo de un grupo de registros especificado (un dominio). Por ejemplo, podemos usar las funciones **DMín** y **DMáx** en los controles calculados de un informe para mostrar el menor y el mayor de los importes de pedido para un cliente particular. O bien, puede utilizar la función **DMín** en una expresión de consulta para mostrar todos los pedidos con un descuento mayor que el descuento mínimo posible.

4.7 Ejemplos expresiones que utilizan funciones de agregado de dominio

4.7.1 Ejemplos generales

Expresión	Descripción
=Dbúsq("[NombreContacto]", "[Proveedores]", "[IdProveedor] = Forms![IdProveedor]")	Utiliza la función Dbúsq (DLookup) para presentar el valor del campo NombreContacto en la tabla Proveedores donde el valor del campo IdProveedor de la tabla se corresponde con el valor del control IdProveedor en el formulario activo.

<p>=Dbúsq("[NombreContacto]", "[Proveedores]", "[IdProveedor] = Forms![Nuevos Proveedores]![IdProveedor]")</p>	<p>Utiliza la función Dbúsq (DLookup) para presentar el valor del campo NombreContacto en la tabla Proveedores donde el valor del campo IdProveedor de la tabla se corresponde con el valor del control IdProveedor en el formulario Nuevos Proveedores.</p>
<p>=DSuma("[CantidadPedido]", "[Pedidos]", "[IdCliente] = 'RATTC'")</p>	<p>Utiliza la función Dsuma (DSum) para presentar la suma total de los valores del campo CantidadPedido en la tabla Pedidos donde el IdCliente es RATTC.</p>

4.7.2 Ejemplo. Resultado de una función de agregado de dominio

Expresión	Resultado
<p>> (DDesvEst("[Carga]", "Pedidos") + DProm("[Carga]", "Pedidos"))</p>	<p>Para un campo Freight, pedidos para los que el cargo por coste queda por encima de la media más la desviación estándar del cargo por coste</p>
<p>>DProm("[Cantidad]", "Detalles de pedidos")</p>	<p>Para un campo Cantidad, <u>productos</u> cuyo pedido exceda la cantidad de pedido media</p>

4.7.3 Ejemplo I. Crear una suma continua en un formulario. DSuma() con una condición

<http://www.gratiszona.com/trucos/trucos-access/crear-suma-continua-formulario.htm>

Para crear una suma continua en un formulario, la tabla sobre la que está basada debe contener al menos un campo autonumerico .

El siguiente ejemplo utiliza una tabla que contiene tres campos: Contador (Campo Autonumerico), Cantidad (Campo numerico) y Total (Campo Numerico). La tabla se llama Prueba.

Crear un formulario nuevo basado en la tabla Prueba.

Formulario: Formulario1

- OrigendelRegistro: Prueba
- Cuadro de Texto: Contador
- Origen del Control: Contador
- Cuadro de Texto: Cantidad
- Origen del Control: Cantidad
- Cuadro de Texto: Total

Origen del control:

Dsum("[Cantidad]";'Prueba';'[Contador]<=Forms![Formulario1]![Contador]')

La funcion Dsum calcula la suma del campo Cantidad en todos los registros que tengan el contador menor o igual que el registro activo. Esta es la razon por la que se necesita un campo autonumerico.

4.7.4 Ejemplo II. DSuma() con varias condiciones y de parámetro

http://www.lawebdelprogramador.com/foros/Access/1128884-funcion_DSum.html

Contamos con una base de datos llamada PETICIONES y en ella en ella hay los siguientes campos (entre otros):

- Producto (texto)
- Dpto (numérico entero)
- Cantidad (numérico entero)
- FechaPetición (fecha corta)
- FechaCancelación (fecha corta)

Contamos a su vez con un formulario "FRM TOTALES" con los siguientes datos:

- un cuadro de texto independiente "FechaDesde"
- un cuadro de texto independiente "FechaHasta"
- un cuadro de texto independiente "Total1"
- un cuadro de texto independiente "Total2"

Pretendemos que una vez introducida las "FechaDesde" y "FechaHasta", en los siguientes campos muestre el total (SUMA) de los datos solicitados. Es decir, debe mostrar en el campo "Total1" el total de lo solicitado en "Producto" para el "Dpto" X en que la "FechaCancelación" sea NULL (que no exista fecha) y que figure la "FechaPetición" entre las indicadas en el formulario ("FechaDesde" y "FechaHasta"),

La propuesta de solución para este caso sería:

= DSum("Cantidad", "PETICIONES", "Producto = 'FIJOS' AND Dpto = 8442 AND

FechaCancelacion = Null AND FechaPeticion Between # " & FechaDesde & " # AND # " &

FechaHasta & " # ")

5 Anexos y bibliografía

5.1 Principales funciones organizadas por categorías

5.1.1 ActiveX

[CreateObject \(función\)](#)

[GetObject \(función\)](#)

5.1.2 Aplicación

[Comando \(función\)](#)

[Ejecutar \(función\)](#)

5.1.3 Matrices

[Array \(función\)](#)

[Filter \(función\)](#)

[Join \(función\)](#)

[LBound \(función\)](#)

[Split \(función\)](#)

[UBound \(función\)](#)

5.1.4 Conversión

[Asc \(función\)](#)

[Car \(función\)](#)

[Día \(función\)](#)

[EuroConvert \(función\)](#)

[FormatoMoneda \(función\)](#)

[FormatoFechaHora \(función\)](#)

[FormatoNúmero \(función\)](#)

[FormatoPorcentaje \(función\)](#)

[GUIDFromString \(función\)](#)

[Hex \(función\)](#)

[Nz \(función\)](#)

[Oct \(función\)](#)

[Str \(función\)](#)

[StringFromGUID \(función\)](#)

[Funciones de conversión de tipo](#)

[Val \(función\)](#)

5.1.5 Base de datos

[DDE \(función\)](#)

[DDEIniciar \(función\)](#)

[DDEPedido \(función\)](#)

[DDEEnviar \(función\)](#)

[Eval \(función\)](#)

[Partición \(función\)](#)

5.1.6 Fecha/Hora

[Fecha \(función\)](#)

[AgregFecha \(función\)](#)

[DifFecha \(función\)](#)

[ParcFecha \(función\)](#)

[SerieFecha \(función\)](#)

[ValorFecha \(función\)](#)
[Hora \(función\)](#)
[Minuto \(función\)](#)
[Mes \(función\)](#)
[NombreMes \(función\)](#)
[Ahora \(función\)](#)
[Segundo \(función\)](#)
[HoraActual \(función\)](#)
[Cronómetro \(función\)](#)
[SerieHora \(función\)](#)
[ValorHora \(función\)](#)
[DíaSemana \(función\)](#)
[NombreDíaDeLaSemana \(función\)](#)
[Año \(función\)](#)

5.1.7 Agregado de dominio

[DProm \(función\)](#)
[DCuenta \(función\)](#)
[DPrim, DÚltimo \(funciones\)](#)
[DBúsq \(función\)](#)
[DMín, DMáx \(funciones\)](#)
[DDesvEst, DDesvEstP \(funciones\)](#)
[DSuma \(función\)](#)
[DVar, DVarP \(funciones\)](#)

5.1.8 Tratamiento de errores

[CVErr \(función\)](#)
[Error \(función\)](#)

5.1.9 Entrada y salida de archivos

[EOF \(función\)](#)
[FreeFile \(función\)](#)
[Entrada \(función\)](#)
[Loc \(función\)](#)
[LOF \(función\)](#)
[Seek \(función\)](#)

5.1.10 Administración de archivos

[DirAct \(función\)](#)
[Dir \(función\)](#)
[FileAttr \(función\)](#)
[HoraFechArch \(función\)](#)
[FileLen \(función\)](#)
[ObtenerAtributo \(función\)](#)

5.1.11 Financieras

[Amortiz \(función\)](#)
[VF \(función\)](#)
[Interés \(función\)](#)
[TIR \(función\)](#)
[TIRM \(función\)](#)
[NPer \(función\)](#)

[VNA \(función\)](#)
[Pago \(función\)](#)
[PPago \(función\)](#)
[VA \(función\)](#)
[Tasa \(función\)](#)
[SLN \(función\)](#)
[SYD \(función\)](#)

5.1.12 *Inspección*

[Entorno \(función\)](#)
[ObtenerValores \(función\)](#)
[ObtenerValor \(función\)](#)
[IsArray \(función\)](#)
[EsFecha \(función\)](#)
[EsVacío \(función\)](#)
[EsError \(función\)](#)
[IsMissing \(función\)](#)
[EsNulo \(función\)](#)
[EsNum \(función\)](#)
[IsObject \(función\)](#)
[TypeName \(función\)](#)
[VarType \(función\)](#)

5.1.13 *Matemáticas*

[Abs \(función\)](#)
[ArcTg \(función\)](#)
[Cos \(función\)](#)
[Exp \(función\)](#)
[Int, Fix \(funciones\)](#)
[Ln \(función\)](#)
[NúmAleat \(función\)](#)
[Redondear \(función\)](#)
[Signo \(función\)](#)
[Sen \(función\)](#)
[Raíz2 \(función\)](#)
[Tan \(función\)](#)

5.1.14 *Mensajes*

[CuadroEntr \(función\)](#)
[CuadroMsj \(función\)](#)

5.1.15 *Varios*

[CallByName \(función\)](#)
[IMEStatus \(función\)](#)
[MacID \(función\)](#)
[MacScript \(función\)](#)
[QBColor \(función\)](#)
[RGB \(función\)](#)
[Spc \(función\)](#)
[Tab \(función\)](#)

5.1.16 Flujo de programas

[Elegir \(función\)](#)

[DoEvents \(función\)](#)

[Silnm \(función\)](#)

[Conmutador \(función\)](#)

5.1.17 Texto

[Formato \(función\)](#)

[EnCad \(función\)](#)

[InStrRev \(función\)](#)

[Minús \(función\)](#)

[Izq. \(función\)](#)

[Longitud \(función\)](#)

[RecortarIzq, RecortarDer y Recortar \(funciones\)](#)

[Medio \(función\)](#)

[Reemplazar \(función\)](#)

[Der \(función\)](#)

[Espacio \(función\)](#)

[CompCadena \(función\)](#)

[ConvCadena \(función\)](#)

[Cadena \(función\)](#)

[StrReverse \(función\)](#)

[Mayús \(función\)](#)

5.2 Bibliografía

<http://office.microsoft.com/es-es/access-help/ejemplos-de-criterios-de-consulta-HA010341674.aspx>
<http://office.microsoft.com/es-es/access-help/informacion-de-referencia-sobre-los-caracteres-comodin-en-access-HA010076601.aspx?CTT=5&origin=HA010341674>
<http://5incoaprendices.blogspot.com.es/2008/07/funcin-diffecha-en-access.html>
<http://office.microsoft.com/es-es/access-help/agregar-y-personalizar-formatos-de-fecha-y-hora-HA010078108.aspx>
<http://accessjuancots.blogspot.com.es/2008/01/capitulo-4-funciones-agregadas-de.html>
<http://www.accessyexcel.com/donde-y-como-utilizar-la-funcion-de-dominio-dlookup/>
<http://office.microsoft.com/es-es/access-help/funciones-dmin-dmin-y-dmax-dmax-HA001228826.aspx>
<http://www.monografias.com/trabajos34/microsoft-access/microsoft-access4.shtml>
<http://josenrique.es/2011/06/separar-nombre-y-apellidos-en-una-tabla-de-personas/>
<http://josenrique.es/2011/06/funciones-de-texto-o-cadena-en-access/>
http://www.lawebdelprogramador.com/foros/Access/112884-funcion_DSum.html