



# ANEXO 1


## Herramienta para facilitar el diseño de consultas.



## El Generador de Expresiones

Se trata de un elemento muy útil para escribir expresiones sin necesidad de conocer la notación que utiliza Access para denominar a sus objetos.

Para activarlo hay que hacer un clic derecho sobre el lugar donde iría la fórmula, y del menú contextual elegir la opción

**Generar....** o bien hacer clic sobre el botón de la varita mágica de la barra de botones.  En ese momento aparecerá la ventana:



*Fig.1 Generador de expresiones*

El Generador de expresiones tiene las secciones siguientes.

**1 Cuadro de expresiones** En la sección superior del generador se encuentra un cuadro en el que se genera la expresión. Utilice la sección inferior para crear los elementos de la expresión, y luego insértelos en el cuadro para formarla. También puede escribir componentes de la expresión directamente en este cuadro.

**2 Botones de operadores** En la sección intermedia del generador hay botones para los operadores que se utilizan con más frecuencia. Al hacer clic en uno de ellos, el Generador de expresiones insertará el operador en el lugar del cuadro de expresiones donde se encuentre el punto de inserción. Para ver una lista completa de los operadores que se pueden emplear haga clic en la carpeta **Operadores** situada en el cuadro inferior izquierdo.

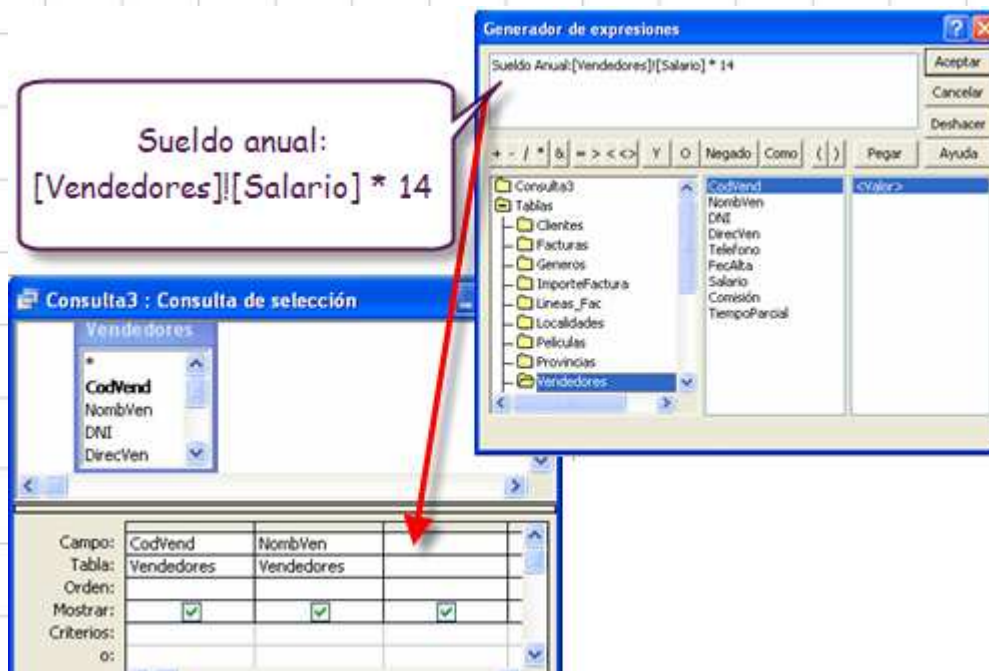
**Elementos de expresiones** En la sección inferior del generador hay tres cuadros:

**3** El cuadro de la izquierda contiene carpetas que muestran los objetos tabla, consulta, formulario e informe de la base de datos, las funciones integradas y definidas por el usuario, las **constantes**, los operadores y las expresiones comunes.

**4** El cuadro intermedio muestra los elementos específicos o las categorías de elementos que corresponden a la carpeta seleccionada en el cuadro de la izquierda. Por ejemplo, al hacer clic en **Funciones incorporadas** en el cuadro de la izquierda, en el cuadro intermedio aparecen las categorías de funciones de Microsoft Access.

5 En el cuadro de la derecha aparecen los valores que corresponden a los elementos seleccionados en los cuadros izquierdo e intermedio. Por ejemplo, al hacer clic en **Funciones incorporadas** en el cuadro de la izquierda y una categoría de funciones en el cuadro intermedio, en el cuadro de la derecha aparecerán todas las funciones incorporadas de la categoría seleccionada.

Por ejemplo imagine que tiene que calcular el sueldo anual de sus vendedores. Para ello tendrá que crear partir de la tabla de Vendedores y crear una consulta en la que pondremos el código, el nombre de los vendedores y una tercera columna donde habrá que multiplicar el sueldo mensual (que es el de la tabla) por 14 (12 meses y 2 pagas). Esto tan simple puede resultar engorroso debido a la sintaxis que utiliza Access en sus expresiones, por ello vamos a usar el generador de expresiones.



## ANEXO 2. Operadores

### Concatenación de texto &

Ejemplo "Hola " & "Mundo" = "Hola Mundo"

---

### Comparación

Operador	Verdadero si	Falso si	Nulo si
< (Menor que)	$expresión1 < expresión2$	$expresión1 >= expresión2$	$expresión1$ o $expresión2 = \text{Null}$
<= (Menor o igual que)	$expresión1 <= expresión2$	$expresión1 > expresión2$	$expresión1$ o $expresión2 = \text{Null}$
> (Mayor que)	$expresión1 > expresión2$	$expresión1 <= expresión2$	$expresión1$ o $expresión2 = \text{Null}$
>= (Mayor o igual que)	$expresión1 >= expresión2$	$expresión1 < expresión2$	$expresión1$ o $expresión2 = \text{Null}$
= (Igual a)	$expresión1 = expresión2$	$expresión1 <> expresión2$	$expresión1$ o $expresión2 = \text{Null}$
<> (Distinto de)	$expresión1 <> expresión2$	$expresión1 = expresión2$	$expresión1$ o $expresión2 = \text{Null}$

---

**EsNulo** Devuelve un valor de tipo **Booleano** (Verdadero o falso) que indica si una expresión contiene datos nulos o no.

**Sintaxis:** **EsNulo** (*expresión*)

El argumento obligatorio *expresión* contiene una expresión numérica o una expresión de cadena.

---

**Como** . Se utiliza para comparar dos cadenas de caracteres.

**Sintaxis:** *resultado* = *cadena* **Como** *patrón*

La sintaxis del operador **Como** consta de las siguientes partes:

Parte	Descripción
<i>resultado</i>	Requerido; cualquier variable numérica.
<i>cadena</i>	Requerido; cualquier expresión de cadena
<i>patrón</i>	Requerido; cualquier expresión de cadena que satisface las convenciones de coincidencia de patrones descritas a continuación.

Caracteres en <i>patrón</i>	Coincidencias en <i>cadena</i>
?	Un carácter cualquiera.
*	Cero o más caracteres.
#	Un dígito cualquiera (0–9).
[ <i>listacaracteres</i> ]	Un carácter cualquiera de <i>listacaracteres</i> .
[! <i>listacaracteres</i> ]	Un carácter cualquiera no incluido en <i>listacaracteres</i> .

---

Entre

**Sintaxis:** **Entre** (*expresion1*) **Y** (*expresión2*)

---

**SiInm** Devuelve uno de entre dos valores, dependiendo de la evaluación de una expresión.

**Sintaxis:** **SiInm** (*expresión, verdadero, falso*)

La sintaxis de la función **SiInm** cuenta con los siguientes argumentos:

Parte	Descripción
<i>expresión</i>	Requerido. Expresión que se desea evaluar.
<i>verdadero</i>	Requerido. Valor o expresión devuelta si <i>expresión</i> es <b>Verdadero</b> .
<i>falso</i>	Requerido. Valor o expresión devuelta si <i>expresión</i> es <b>Falso</b> .

## ANEXO 3. Funciones de cadena de texto

**Longitud** Devuelve el número de caracteres que hay en una cadena de texto

**Sintaxis:** **Longitud** (*cadena* | *nombrevar*)

Como parámetro se pone bien una cadena de texto entre comillas dobles, o bien el nombre de una variable (por ejemplo: un campo de una tabla). Si *cadena* o *nombrevar* contiene nulo la función devuelve nulo.

---

**Mayús** Devuelve la cadena de texto en mayúsculas

**Sintaxis:** **Mayús** (*Cadena*)

---

**Minús** Devuelve la cadena de texto en minúsculas

**Sintaxis:** **Minús** (*Cadena*)

---

**EnCad** Devuelve un número que indica la posición de la primera aparición de *cadena1* en *cadena2*.

**Sintaxis:** **EnCad** ([*inicio*]; *Cadena1*; *Cadena2* [: *comparar*])

La sintaxis de la función **EnCad** tiene estos argumentos:

Argumento	Descripción
<i>inicio</i>	Opcional. Número que indica la posición inicial para cada búsqueda. Si se omite, la búsqueda comienza en la posición del primer carácter. Si <i>inicio</i> contiene un valor Nulo, se produce un error. El argumento <i>inicio</i> se requiere si se especifica <i>comparar</i> .
<i>Cadena1</i>	Obligatorio. Cadena de texto en la que se busca.
<i>Cadena2</i>	Obligatorio. Cadena de texto buscada.
<i>comparar</i>	Opcional. Especifica el tipo de comparación de cadena. Si <i>comparar</i> es Nulo, ocurre un error. Vea la tabla <i>comparar</i> al final.

### Valores devueltos

Si	La función EnCad devuelve
<i>Cadena1</i> es de longitud cero	0
<i>Cadena1</i> es nulo	Nulo
<i>Cadena2</i> es de longitud cero	<i>inicio</i>
<i>Cadena2</i> es nulo	Nulo
<i>Cadena2</i> no se encontró	0
<i>Cadena2</i> se encontró dentro de <i>Cadena1</i>	Posición en la que se halla la coincidencia
<i>inicio</i> > <i>Cadena2</i>	0

---

**InStrRev** Devuelve la posición de una aparición de una cadena dentro de otra, contando desde la derecha de la cadena

**Sintaxis:** **InstrRev(cadena1, cadena2[, inicio[, comparar]])**

La sintaxis de la función **InstrRev** consta de los siguientes argumentos.

Argumento	Descripción
<i>cadena1</i>	Obligatorio: Cadena de texto en la que se busca.
<i>cadena2</i>	Obligatorio. Cadena de texto que se busca.
<i>inicio</i>	Opcional. Número que establece la posición inicial para cada búsqueda. Si se omite, se utiliza -1, que significa que la búsqueda empieza en la posición del último carácter. Si el inicio contiene nulo, se produce un error.
<i>comparar</i>	Opcional. Un valor numérico que indica la clase de comparación que se utilizará cuando se evalúen subcadenas. Si se omite, se realiza una comparación binaria. Vea la sección Valores para consultar los valores posibles.

**InStrRev** devuelve los valores siguientes:

Si	InStrRev devuelve
<i>cadena1</i> es de longitud cero	0
<i>cadena1</i> es Null	Null
<i>cadena2</i> es de longitud cero	<i>Inicio</i>
<i>cadena2</i> es Null	Null
<i>cadena2</i> no se encuentra	0
<i>cadena2</i> se encuentra dentro de <i>cadena1</i>	La posición en la que se encuentra la coincidencia
<i>inicio</i> > Len( <i>cadena2</i> )	0

**Medio**. Devuelve una cadena de texto de una longitud determinada

**Sintaxis:** **Medio (Cadena; inicio; longitud)**

La sintaxis de la función **Medio** tiene estos argumentos con nombre:

Argumento	Descripción
<b>Cadena</b>	Obligatorio. Cadena de texto de la cual se devuelven los caracteres. Si <b>cadena</b> contiene nulo, se devuelve <b>Nulo</b> .
<b>inicio</b>	Obligatorio; Posición de carácter en <b>Cadena</b> en la cual comienza el texto que se quiere extraer. Si <b>inicio</b> es mayor que el número de caracteres en la <b>Cadena</b> , <b>Mid</b> devuelve una cadena de longitud cero ("").
<b>longitud</b>	Obligatorio; Número de caracteres que se van a devolver. Si se omite o en el texto hay menos de <b>longitud</b> caracteres (incluyendo el carácter de <b>inicio</b> ), se devuelven todos los caracteres desde la posición de <b>inicio</b> hasta el final de la cadena.

**Der**. Devuelve un número especificado de caracteres de una cadena comenzando por la derecha.

**Sintaxis:** **Der (Cadena; n)**

La sintaxis de la función **Right** tiene estos argumentos

Argumento	Descripción
<b>cadena</b>	Obligatorio. Cadena desde la cual se devuelven los caracteres que están más a la derecha. Si <b>cadena</b> contiene nulo, se devuelve <b>Nulo</b> .
<b>n</b>	Obligatorio, Número que indica cuántos caracteres se van a devolver. Si es 0, se devuelve una cadena de longitud cero (""). Si es mayor o igual al número de caracteres en <b>cadena</b> , se devuelve la cadena completa.



**Izq** Devuelve un número especificado de caracteres de una cadena comenzando por la izquierda.

**Sintaxis** **Izq (Cadena; n)**

La sintaxis de la función **Izq** tiene estos argumentos con nombre:

Argumento	Descripción
<b>cadena</b>	Obligatorio. Cadena de la cual se devuelven los caracteres que están más a la izquierda. Si <b>cadena</b> contiene Nulo, se devuelve <b>Nulo</b> .
<b>n</b>	Obligatorio, Número que indica cuántos caracteres se van a devolver. Si es 0, devuelve una cadena de longitud cero (""). Si es mayor o igual al número de caracteres en <b>cadena</b> , se devuelve la cadena entera.

**RecortarIzq** Devuelve una copia de una cadena determinada sin espacios a la izquierda

**RecortarDer** Devuelve una copia de una cadena determinada sin espacios a la derecha

**Recortar** Devuelve una copia de una cadena determinada sin espacios ni a la derecha ni a la izquierda

**Sintaxis** **RecortarIzq (Cadena)** **RecortarDer (Cadena)** **Recortar (Cadena)**

El argumento **cadena** es cualquier cadena de texto válida. Si **cadena** contiene **Nulo**, se devuelve **Nulo**

**CompCadena** Devuelve el resultado de una comparación de dos cadenas de texto

**Sintaxis** **CompCadena (Cadena1; Cadena2; comparar]**

La sintaxis de la función **StrComp** tiene estos argumentos:

Argumento	Descripción
<b>Cadena1</b>	Obligatorio. Cualquier cadena de texto válida.
<b>Cadena2</b>	Obligatorio. Cualquier cadena de texto válida.
<b>comparar</b>	Opcional. Especifica el tipo de comparación de cadena. Si el argumento <b>comparar</b> es nulo el valor de <b>Option Compare</b> determina el tipo de comparación.

La función **CompCadena** devuelve los siguientes valores:

Si	La función StrComp devuelve
<b>Cadena1</b> es menor que <b>cadena2</b>	-1
<b>Cadena1</b> es igual a <b>cadena2</b>	0
<b>Cadena1</b> es mayor que <b>cadena2</b>	1
<b>Cadena1</b> o <b>cadena2</b> es <b>Nulo</b>	<b>Nulo</b>

**StrReverse** Devuelve una cadena en la que se invierte el orden de carácter de una cadena especificada.

**Sintaxis** **StrReverse(cadena1)**

El argumento **cadena1** es la cadena cuyos caracteres se invierten. Si **cadena1** es una cadena de longitud cero (""), se devuelve una cadena de longitud cero. Si es **Nulo**, se produce un error.

### Valores

Los valores del argumento **comparar** son:

Constante	Valor	Descripción
<b>vbUseCompareOption</b>	-1	Realiza una comparación con los valores de la instrucción <b>Option Compare</b> .
<b>vbBinaryCompare</b>	0	Realiza una comparación binaria.
<b>vbTextCompare</b>	1	Realiza una comparación textual.
<b>vbDatabaseCompare</b>	2	Sólo para Microsoft Access. Realiza una comparación basada en la información en las bases de datos.





## ANEXO 4. Funciones de fecha

**Fecha()** Devuelve la fecha actual del sistema.

**Sintaxis:** *Fecha()*

---

**Ahora()** Devuelve la fecha y hora actuales según la configuración de la fecha y la hora del sistema de su equipo.

**Sintaxis:** *Ahora()*

---

**Día** Devuelve un número entero entre 1 y 31, inclusive, que representa el día del mes.

**Sintaxis:** *Día(*fecha*)*

Si *fecha* contiene Nulo la función devolverá **Nulo**.

---

**Mes** Devuelve un número entero entre 1 y 12, inclusive, que representa el mes del año.

**Sintaxis:** *Mes(*fecha*)*

El argumento *fecha* es cualquier expresión que pueda representar una fecha. Si *fecha* contiene Nulo la función devolverá **Nulo**.

---

**Año** Devuelve un número entero que representa el año.

**Sintaxis:** *Año(*fecha*)*

El argumento *fecha* es cualquier expresión que puede representar una fecha. Si *fecha* contiene Null, la función devolverá **Null**.

---

**DíaSemana** Devuelve un número entero, del 1 al 7, que representa el día de la semana.

**Sintaxis:** *DíaSemana(*fecha*, [*primerdíasemana*])*

La sintaxis de la función **DíaSemana** consta de los siguientes argumentos

Parte	Descripción
<i>fecha</i>	Obligatorio. Una fecha. Si <i>fecha</i> contiene Nulo, la función devolverá <b>Nulo</b>
<i>primerdíasemana</i>	Opcional. Una constante que especifica el primer día de la semana. Si no se especifica, se asume el Lunes. Vea la tabla <i>primerdíasemana</i> al final.

**Ejemplo.** Número del día: *DíaSemana([Facturas],[Fecha])*

---

**ParcFecha** Devuelve la parte especificada de fecha dada. Esto es el día, el mes, el año, etc..

**Sintaxis: ParcFecha(intervalo, fecha[, primerdíasemana[, primerasemanaaño]])**

La sintaxis de la función **DatePart** consta de los siguientes argumentos con nombre:

Parte	Descripción
<i>intervalo</i>	Obligatorio. Expresión de cadena con el intervalo de tiempo que desea devolver. Vea la tabla <i>intervalo</i> al final.
<i>fecha</i>	Obligatorio. Fecha que se desea evaluar.
<i>primerdíasemana</i>	Opcional. Una constante que especifica el primer día de la semana. Si no se especifica, se asume que es el Lunes. Vea la tabla <i>primerdíasemana</i> al final.
<i>primerasemanaaño</i>	Opcional. Una constante que especifica la primera semana del año. Si no se especifica, se asume que es aquella en la que se encuentre el 1 de enero. Vea la tabla <i>primerasemanaaño</i> al final.

**NombreDíaDeLaSemana** Devuelve una cadena que indica el día de la semana especificado.

**Sintaxis: NombreDíaDeLaSemana(díaDeLaSemana [, abreviar] [, primerDíaDeLaSemana])**

La sintaxis de la función **DíaSemana** consta de las siguientes partes:

Parte	Descripción
<i>díaDeLaSemana</i>	Obligatorio. Número del 1 al 7 para el día de la semana. El valor numérico de cada día depende de la configuración del valor <i>primerdíasemana</i> .
<i>abreviar</i>	Opcional. Falso: devuelve el nombre completo del día. Verdadero: devuelve las tres primeras letras del día. Si se omite, el valor predeterminado es <b>Falso</b> .
<i>primerDíaDeLaSemana</i>	Opcional. Un valor numérico que indica el primer día de la semana. Vea la tabla <i>primerasemanaaño</i> al final.

**Ejemplo Nombre del día:** NombreDíaDeLaSemana(DíaSemana([Facturas]![Fecha]))

**NombreMes** Devuelve una cadena que indica el mes especificado.

**Sintaxis: NombreMes(mes [, abreviar])**

La sintaxis de la función **MonthName** consta de las siguientes partes:

Parte	Descripción
<i>mes</i>	Obligatorio. La designación numérica del mes. Por ejemplo, enero es 1, febrero es 2, etcétera.
<i>abreviar</i>	Opcional. Falso: devuelve el nombre completo del mes. Verdadero: devuelve las tres primeras letras del mes. Si se omite, el valor predeterminado es <b>Falso</b> .

**AgregFecha** Devuelve una fecha a la que se ha agregado un intervalo de tiempo especificado.

**Sintaxis: AgregFecha(intervalo, número, fecha)**

La sintaxis de la función **AgregFecha** consta de los siguientes argumentos con nombre:

Parte	Descripción
<i>intervalo</i>	Obligatorio. Expresión de cadena que contiene el intervalo de tiempo que se desea agregar. Vea la tabla <i>intervalo</i> , más abajo
<i>número</i>	Obligatorio. Expresión numérica con el número de intervalos que se desea agregar. Puede ser positiva (para obtener fechas futuras) o negativa (para obtener fechas pasadas).
<i>fecha</i>	Obligatorio. Valor que representa la fecha en la que se agregó el intervalo.



**DiffFecha** Devuelve el número de intervalos de tiempo entre dos fechas determinadas.

**Sintaxis:** **DiffFecha(intervalo, fecha1, fecha2[, primerdíasemana[, primerasemanaaño]])**

La sintaxis de la función **DiffFecha** consta de los siguientes argumentos:

Parte	Descripción
<b>Intervalo</b>	Obligatorio. Expresión con el intervalo de tiempo utilizado para calcular la diferencia entre <b>fecha1</b> y <b>fecha2</b> .. Vea tabla <b>intervalo</b> más abajo.
<b>Fecha1, fecha2</b>	Obligatorio: Las dos fechas que se van a utilizar en el cálculo.
<b>Primerdíasemana</b>	Opcional. Constante que especifica el primer día de la semana. Si no se especifica, se asume que es el lunes. Vea la tabla <b>primerdíasemana</b> .
<b>Primerasemanaaño</b>	Opcional. Constante, que especifica la primera semana del año. Si no se especifica, se asume que es aquella en la que se encuentre el 1 de enero.

## TABLAS

El argumento **intervalo** toma los siguientes valores:

Valor	Descripción
aaaa	Año
t	Trimestre
m	Mes
a	Día del año
d	Día
e	Día de la semana
ee	Semana
h	Hora
n	Minuto
s	Segundo

El argumento **primerdíasemana** toma los siguientes valores:

Constante	Valor	Descripción
<b>vbUseSystem</b>	0	Utiliza el valor de la API de NLS.
<b>vbSunday</b>	1	Domingo (predeterminado)
<b>vbMonday</b>	2	Lunes
<b>vbTuesday</b>	3	Martes
<b>vbWednesday</b>	4	Miércoles
<b>vbThursday</b>	5	Jueves
<b>vbFriday</b>	6	Viernes
<b>vbSaturday</b>	7	Sábado

El argumento **primerasemanaaño** toma los siguientes valores:

Constante	Valor	Descripción
<b>vbUseSystem</b>	0	Utiliza el valor de la API de NLS.
<b>vbFirstJan1</b>	1	Empieza con la semana en la que se encuentra el 1 de enero (predeterminado).
<b>vbFirstFourDays</b>	2	Empieza con la primera semana que tenga al menos cuatro días en el nuevo año.
<b>vbFirstFullWeek</b>	3	Empieza con la primera semana que esté completamente incluida en el nuevo año.