

Las fechas en VBA

Tras muchos años trabajando con datos, todos ellos asociados a fechas, he llegado a la conclusión de que tanto que se dice de ellas no tiene justificación.

Al comienzo tuve problemas por desconocimiento, temporalmente los resolví utilizando esas fechas en campos y variables de texto, pero esa tampoco era la solución, cuando alguien, no recuerdo si el Búho, Eduardo Oláz o quizá Chea, me dijo que las utilizará como tales y que en SQL debía usar el formato americano "mm/dd/yy", se acabaron mis problemas.

Bueno, casi, por entonces solo hacía cierres mensuales de datos, en el momento que tuve que realizar cierres semanales y anuales, las complicaciones empezaron de nuevo, todo porque la función **Format** devuelve un dato de semana incorrecto en algunas fechas de primero o fin de año.

Salí del paso creando una tabla calendario en la que inserté todas las fechas, desglosando cada una de sus partes y corrigiendo manualmente aquellas que no eran correctas.

Después supe que la [norma ISO 8601](#) trata de resolver los problemas asociados a las fechas, como por ejemplo que un año tenga semanas de 6, 8 o incluso un día.

Establece que la semana 1 de cada año es aquella que contiene cuatro días de ese año, o lo que es lo mismo, la que contiene el primer jueves del año.

Según eso, un día de un año puede pertenecer a una semana de otro año, y esa era mi cruz, si un día pertenece a la semana 1 del año anterior, y a su año correspondiente, al unir ambos datos resulta un desastre.

Un ejemplo:

El 31 de diciembre de 2018, lunes, semana 1 (de 2019) sin embargo su año es 2018, por lo que sus datos acabarían mezclados con los de la semana 1 de 2018 y en consecuencia el resultado, una calamidad.

La norma resuelve ese problema, ya que dice que las semanas deben representarse en formato **aaaaWww** (año en cuatro cifras, seguidos de la letra W en mayúsculas y del número de semana, con dos cifras), en el ejemplo que nos ocupa la fecha 31/12/2018 pertenece a la semana **2019W01**.

Siendo así la semana 2018W01 tendrá 7 días, no 8 como habría tenido mientras que la 2019W01 perdería uno y tendría 6.

Dicho sea de paso, según la misma norma, la fecha se debe representar en formato **aaaa-mm-dd** y si incluyera una hora se añadiría una T mayúscula y continuaría con el formato clásico **hh:mm:ss: aaaa-mm-ddThh:mm:ss**.

Ese formato de fecha garantiza que el orden alfabético de las fechas siempre sea correcto.

La siguiente tabla resume los formatos establecidos por la norma

Las fechas en VBA

Formatos para escribir Fechas y Horas según la norma ISO 8601		
Fecha u Horas	Formato ISO 8601	Ejemplo Referidos al Jueves 24/11/2017, semana 47
Fecha	yyyy-mm-dd	2017-11-24
Hora	hh:mm:ss.ffffff	19:17:00.322348
Hora con zona horaria	hh:mm:ss.ffffff+ -hh:mm	19:17:00+02:00
Fecha y hora	yyyy-mm-ddThh:mm:ss.ffffff	2017-11-24T24:17:00
Fecha y hora local con zona horaria	yyyy-mm-ddThh:mm:ss.ffffff+ -hh:mm	2017-11-24T24:17:00+02:00
Número de semana	yyyyWww	2017W47
Número de semana completo	yyyyWwwd	2017W475

Donde esos dígitos representan:

y	año
m	mes
d	día
h	hora
m	minuto
s	segundo
f	fracción de segundo
w	semana

Después de todas estas peripecias llego a las siguientes conclusiones:

1. Las fechas no muerden.
2. Trabaja siempre con fechas, no hay necesidad de hacerlo con su equivalente numérico.
3. En SQL, utiliza siempre el formato mm/dd/yyyy.
4. Cuando trabajes con variables fechas, estas no soportan nulos y su contenido hasta su asignación es cero.
5. Crea una tabla Calendario y asíciala al resto de tus tablas, no consumas recursos para calcular el mes, año, semana, etc, utiliza los de la tabla.

tblCalendario											
Fecha	Año	Semana	Año	Trimestre	Mes	Semana	Día	Día	Semana	Fecha ISO8601	Semana ISO8601
28/12/2018	2018	2018	4	12	52	28	5	2018-12-28	2018W52		
29/12/2018	2018	2018	4	12	52	29	6	2018-12-29	2018W52		
30/12/2018	2018	2018	4	12	52	30	7	2018-12-30	2018W52		
31/12/2018	2019	2018	4	12	1	31	1	2018-12-31	2019W01		
01/01/2019	2019	2019	1	1	1	1	2	2019-01-01	2019W01		
02/01/2019	2019	2019	1	1	1	2	3	2019-01-02	2019W01		
03/01/2019	2019	2019	1	1	1	3	4	2019-01-03	2019W01		

Los siguientes procedimientos crean esa tabla "Calendario" con fechas desde 1900 hasta 2050, si necesitas mas seguro que sabes como resolverlo.

Las fechas en VBA

```
*****
'* CrearTablaCalendario
'* crea una tabla e incluye en ella todos los datos de las fechas entre
'* 01/01/1900 y 31/12/2050
'* uso: CrearTablaCalendario
'* http://www.mvp-access.es/emilio
'* Si utilizas este código, respeta la autoría y los créditos
'* ESH 19/11/17 13:28
*****

Public Sub CrearTablaCalendario()
Dim strSQL As String, _
    datFecha As Date, _
    strSemana As String
Dim Comando As Object 'ADODB.Command
Dim Catalogo As Object 'ADOX.Catalog

DoCmd.Hourglass True

' borro objetos relacionados
On Error Resume Next
DoCmd.DeleteObject acTable, "tblCalendario"
DoCmd.DeleteObject acQuery, "qryVerificarSemanas"
DoCmd.DeleteObject acQuery, "qryDiferencias"

On Error GoTo CrearTablaCalendario_TratamientoErrores

' creo la tabla tblCalendario
strSQL = "CREATE TABLE tblCalendario"
strSQL = strSQL & " (id Long CONSTRAINT PrimaryKey PRIMARY KEY,"
strSQL = strSQL & " [Fecha] Date,"
strSQL = strSQL & " [AñoSemana] Short,"
strSQL = strSQL & " [Año] Short,"
strSQL = strSQL & " [Trimestre] Byte,"
strSQL = strSQL & " [Mes] Byte,"
strSQL = strSQL & " [Semana] Byte,"
strSQL = strSQL & " [Dia] Byte,"
strSQL = strSQL & " [DiaSemana] Byte,"
strSQL = strSQL & " [Fecha ISO8601] Text(10),"
strSQL = strSQL & " [Semana ISO8601] Text(7),"
strSQL = strSQL & " [Semana Format] Byte)"
DBEngine(0)(0).Execute strSQL, dbFailOnError

' convierto el campo id en autonumérico porque DAO, mediante SQL, no me deja crearlo directamente como tal
strSQL = "ALTER TABLE tblCalendario ALTER COLUMN ID AutoIncrement"
CurrentDb.Execute strSQL

For datFecha = #1/1/1900# To #12/31/2050# 'DateAdd("yyyy", 1, datMaxFecha) - 1
    strSemana = NumeroSemanaISO8601(datFecha)
    strSQL = "INSERT INTO tblCalendario ([Fecha], AñoSemana, Año, Trimestre, Mes, Semana, Dia, DiaSemana, [Fecha ISO8601], [Semana ISO8601], [Semana Format])"
    strSQL = strSQL & " SELECT " & FechaSQL(datFecha) & ","
    strSQL = strSQL & Left$(strSemana, 4) & ","
    strSQL = strSQL & Year(datFecha) & ","
    strSQL = strSQL & Int((Month(datFecha) - 1) / 3) + 1 & ","
    strSQL = strSQL & Month(datFecha) & ","
    strSQL = strSQL & Right$(strSemana, 2) & ","
    strSQL = strSQL & Day(datFecha) & ","
    strSQL = strSQL & Weekday(datFecha, vbMonday) & ","
    strSQL = strSQL & "" & Format$(datFecha, "yyyy-mm-dd") & ","
    strSQL = strSQL & "" & strSemana & ","
    strSQL = strSQL & Format$(datFecha, "ww", vbMonday, vbFirstFourDays)
    DBEngine(0)(0).Execute strSQL, dbFailOnError
Next datFecha

' creo una consulta para mostrar las semanas que tengan mas o menos de 7 días
Set Catalogo = CreateObject("ADOX.Catalog")
' conexión
Catalogo.ActiveConnection = CurrentProject.Connection

MsgBox "Vamos a verificar si todas las semanas tienen 7 días", vbYes
strSQL = "SELECT [Fecha], Semana, [Semana ISO8601], DiaSemana"
strSQL = strSQL & " FROM tblCalendario"
strSQL = strSQL & " WHERE [Semana ISO8601]"
strSQL = strSQL & " IN ("
strSQL = strSQL & " SELECT [Semana ISO8601]"
strSQL = strSQL & " FROM tblCalendario"
strSQL = strSQL & " GROUP BY Semana, [Semana ISO8601]"
strSQL = strSQL & " HAVING NOT Count([Semana ISO8601]) = 7"
strSQL = strSQL & " )"
strSQL = strSQL & " ORDER BY [Fecha]"
Set Comando = CreateObject("ADODB.Command")
Comando.CommandText = strSQL

MsgBox "Solo salen los 6 últimos días de 2050 (último año del periodo), semana incompleta luego resultado correcto", vbYes

' creo la consulta
Catalogo.Procedures.Append "qryVerificarSemanas", Comando
' la abro
DoCmd.OpenQuery "qryVerificarSemanas"

MsgBox "Ahora vamos a ver cual es el comportamiento del Format de VBA", vbYes
strSQL = "SELECT Fecha, Semana AS [Semana ISO], [Semana Format]"
```

Las fechas en VBA

```
strSQL = strSQL & " FROM tblCalendario"
strSQL = strSQL & " WHERE NOT [Semana] = [Semana Format]"
Set Comando = CreateObject("ADODB.Command")
Comando.CommandText = strSQL

' creo la consulta
Catalogo.Procedures.Append "qryDiferencias", Comando
' la abro
DoCmd.OpenQuery "qryDiferencias"

MsgBox "Se comprueba que el Format de VBA ocasionalmente devuelve resultados incorrectos", vbYes

If MsgBox("¿Eliminar de la tabla el campo ""Semana Format"" incorrecto?", vbYesNo, "Atención pregunta") = vbYes Then
    DoCmd.Close acQuery, "qryVerificarSemanas"
    DoCmd.Close acQuery, "qryDiferencias"
    strSQL = "ALTER TABLE tblCalendario DROP COLUMN [Semana Format]"
    CurrentDb.Execute strSQL
End If

CrearTablaCalendario_Salir:
    On Error Resume Next
    DoCmd.Hourglass False
    On Error GoTo 0
    Exit Sub

CrearTablaCalendario_TratamientoErrores:
    MsgBox "Error " & Err & ": " & Err.Description & vbNewLine & Switch(Erl = 0, vbNullString, Not Erl = 0, "En línea: " & Erl) & ",
del procedimiento: CrearTablaCalendario, del Módulo: " & Application.VBE.ActiveCodePane.CodeModule.Name, vbCritical + vbOKOnly,
DBEngine(0)(0).Properties("AppTitle")
    Resume CrearTablaCalendario_Salir

End Sub

'*****
'* FechaSQL
'* devuelve una cadena debidamente formateada para su uso en sentencias SQL, es
'* decir con formato mm/dd/yy y, si fuera preciso, hh:nn:ss
'* Argumentos: datFecha => fecha a formatear
'* uso: FechaSQL (Now) devuelve #08/25/15 22:45:36#
'* ESH 25/08/15 22:45
'*****

Public Function FechaSQL(datFecha As Date) As String
' según la hora sea entera o no, aplicaré un formato u otro
If Int(datFecha) - datFecha = 0 Then
    FechaSQL = Format$(datFecha, "\#mm\dd\yyyy#")
Else
    FechaSQL = Format$(datFecha, "\#mm\dd\yyyy hh\:nn\:ss#")
End If
End Function

'*****
'* NumeroSemanaISO8601
'* Devuelve la semana de la fecha pasada o por defecto de la actual según la
'* norma ISO 8601, que dice que la primera semana del año es la que tiene al
'* menos cuatro días del nuevo año o lo que es lo mismo aquella que contiene el
'* primer jueves del año
'* Argumentos: datFecha => (Opcional) Fecha de calculo
'* blnCompleta => (Opcional) devolver en formato YYYYW00 (True)
'* devolver en formato 00 (False)
'* uso: NumeroSemanaISO8601(Date)
'* ESH 17/11/17 16:05
'*****

Public Function NumeroSemanaISO8601(Optional datFecha As Date, Optional blnCompleta As Boolean = True) As String
Dim datJueves As Date

On Error GoTo NumeroSemanaISO8601_TratamientoErrores

If datFecha = 0 Then datFecha = Date
' busco la fecha del jueves de esa semana
datJueves = (datFecha - Weekday(datFecha, vbMonday) + 4)
If blnCompleta Then
    ' devuelvo el año y número de semana de ese jueves, concatenados con una W para cumplir con la ISO8601
    NumeroSemanaISO8601 = Year(datJueves) & "W" & Format$(Format$(datJueves, "ww", vbMonday, vbFirstFourDays), "00")
Else
    NumeroSemanaISO8601 = Format$(Format$(datJueves, "ww", vbMonday, vbFirstFourDays), "00")
End If

NumeroSemanaISO8601_Salir:
    On Error GoTo 0
    Exit Function

NumeroSemanaISO8601_TratamientoErrores:
    MsgBox "Error " & Err & ": " & Err.Description & vbNewLine & Switch(Erl = 0, vbNullString, Not Erl = 0, "En línea: " & Erl) & ",
)" & "del procedimiento: NumeroSemanaISO8601, del Módulo: " & Application.VBE.ActiveCodePane.CodeModule.Name, vbCritical + vbOKOnly
    Resume NumeroSemanaISO8601_Salir

End Function
```