|  |
| --- |
| [logo_gj01.gif](htto://joaquin.medina.name/) |
| Apuntes Tácticos |
| El Formato de Fecha Week Date  (ISO 8601:2004) |

**Información del documento**

|  |  |
| --- | --- |
|  | La Güeb de Joaquin (<http://joaquin.medina.name>) |
| Categoría | Programación .NET |
| Tema.: | Formatos de Fecha |
| Titulo | El Formato de Fecha Week Date (ISO 8601:2004) |
| Autor.: | Joaquin Medina Serrano ([mailto:joaquin@medina.name](mailto:joaquin@medina.name?subject=Documento%20Patrón%20de%20comportamiento%20Mediador)) |
| Archivo | Informatica/Lenguajes/.NET/System/Date |
| Sumario | La norma Internacional ISO 8601:2004 específica las representaciones numéricas de la fecha y hora. Esta notación estándar ayuda a evitar la confusión en la comunicación internacional causada por las diferentes notaciones nacionales y se aumenta la portabilidad de las interfaces de usuario de la computadora.  En las aplicaciones comerciales e industriales existen muchas actividades que se organizan por semanas, como la rotación de turnos, los plazos de entrega, los planes de producción, etc. a menudo es necesario hacer referencia a una semana de un año, y en estos casos, es más práctico saber el número de semana y el día de la semana que conocer el día del mes. |
| Numero de paginas |  |
| Fecha de creación. |  |
| Fecha última modificación. |  |

**Sobre el autor**

|  |  |
| --- | --- |
| Autobombo del autor | * Joaquín Medina Serrano [[Web](http://joaquin.medina.name)] [ [Mail](mailto:joaquin@medina.name%20) ] * Fotógrafo erótico conceptual * Autor de libros sobre Visual Basic, Cobol y Office * He desarrollado diversas aplicaciones de gestión, utilizando herramientas como Clipper, Visual Basic 6 y Visual Basic .NET. * Autor de artículos sobre Programación Orientada a Objetos, y Visual Basic .NET * El último curso de capacitación realizado ha sido el [Máster en Servicios Web, Seguridad Informática y Aplicaciones de Comercio Electrónico](http://wzar.unizar.es/servicios/epropios/oferta/151.html) de la [[Universidad de Zaragoza](http://wzar.unizar.es/) ] |

**Contenido**

[1 La norma ISO 8601:2004 4](#_Toc253323404)

[1.1 Las Semanas 5](#_Toc253323405)

[1.1.1 Numeración de las semanas 5](#_Toc253323406)

[1.1.2 ¿Puede haber años con menos de una semana? 5](#_Toc253323407)

[1.1.3 ¿Puede haber años con 53 o más semanas? 5](#_Toc253323408)

[1.1.4 ¿Es posible que existan años con 51 semanas o menos?: 5](#_Toc253323409)

[1.2 El Formato de Fecha Week Date 6](#_Toc253323410)

[1.3 Formatos de presentación según la norma ISO 6](#_Toc253323411)

[2 Cálculos diversos 6](#_Toc253323412)

[2.1 Averiguar el número de semanas de un año 6](#_Toc253323413)

[2.2 Averiguar el número de semanas de un mes 8](#_Toc253323414)

[2.2.1 Código VB.NET 9](#_Toc253323415)

[2.3 Transformar una fecha Gregoriana a Ordinal 10](#_Toc253323416)

[2.4 Transformar una fecha Gregoriana a fecha Week Date 10](#_Toc253323417)

[2.5 Transformar una fecha Week Date a fecha Ordinal 12](#_Toc253323418)

[2.6 Transformar una fecha Week Date a fecha Gregoriana 13](#_Toc253323419)

[2.7 Transformar una fecha Ordinal a fecha Gregoriana 13](#_Toc253323420)

[2.8 Transformar una fecha Ordinal a fecha Week Date 14](#_Toc253323421)

[3 Código VB.NET 15](#_Toc253323422)

[3.1 Averiguar el número de semanas de un mes 15](#_Toc253323423)

[3.2 Averiguar el número de semanas de un año 16](#_Toc253323424)

[3.3 Calculo del número de la semana de una fecha 18](#_Toc253323425)

[3.4 Transformar una fecha Gregoriana a fecha Ordinal 20](#_Toc253323426)

[3.5 Transformar una fecha Ordinal a fecha Gregoriana 23](#_Toc253323427)

[3.6 Transformar una fecha Ordinal a fecha Week Date 25](#_Toc253323428)

[3.7 Transformar una fecha Week Date a fecha Ordinal 28](#_Toc253323429)

[3.8 Transformar una fecha Gregoriana a fecha Week Date 30](#_Toc253323430)

[3.9 Transformar una fecha Week Date a fecha Gregoriana 30](#_Toc253323431)

El Formato de Fecha Week Date

(ISO 8601:2004)

**Sumario**

En las aplicaciones comerciales e industriales existen muchas actividades que se organizan por semanas, como la rotación de turnos, los plazos de entrega, los planes de producción, etc. a menudo es necesario hacer referencia a una semana de un año, y en estos casos, es más práctico saber el número de semana y el día de la semana que conocer el día del mes.

# La norma ISO 8601:2004

La norma Internacional ISO 8601:2004 específica las representaciones numéricas de la fecha y hora. Esta notación estándar ayuda a evitar la confusión en la comunicación internacional causada por las diferentes notaciones nacionales y se aumenta la portabilidad de las interfaces de usuario de la computadora.

<http://es.wikipedia.org/wiki/ISO_8601>

<http://wapedia.mobi/es/ISO_8601>

Las ventajas del estándar ISO 8601 con respecto a otras notaciones son:

* Es fácil procesar estas fechas sin tablas de meses auxiliares.
* Es fácil compararlas y ordenarlas.
* Independiente del idioma.
* No se confunde con otras notaciones.

El estándar internacional para la notación de la fecha Gregoriana es:

Formato básico YYYYMMDD Ejemplo 19850412

Formato extendido YYYY-MM-DD Ejemplo 1985-04-12

Donde YYYY es el año en el calendario gregoriano, MM es el mes en el año entre 01 (Enero) y 12 (Diciembre), y DD es el día del mes entre 01 y 31.

El estándar Internacional para la notación de la fecha por semanas (Week Date y Calendar Week) es:

Completa (un día específico) (Week Date)

Formato básico YYYYWwwD Ejemplo 1985W155

Formato extendido YYYY-Www-D Ejemplo 1985-W15-5

Reducida (una semana específica) (Calendar Week)

Formato básico YYYYWww Ejemplo 1985W15

Formato extendido YYYY-Www Ejemplo 1985-W15

Donde YYYY es el año del calendario gregoriano, W es el carácter separador (e identificador de este tipo de formato) ww es el numero de la semana del año, y D es el día de la semana siendo le lunes =1 y el domingo=7

Y por último la notación Ordinal

Formato básico YYYYDDD Ejemplo 1985102

Formato extendido YYYY-DDD Ejemplo 1985-102

Donde YYYY es el año del calendario gregoriano, y DDD es el número del día dentro del año, contando a partir del uno de enero del año

La norma también define el año Bisiesto como los años que tienen un día adicional en febrero (YYYY-02-29), y son aquellos cuyo año es un múltiplo de 4 con las siguientes excepciones:

* Si el año es un múltiplo de 100, sólo es bisiesto ese año si es a su vez múltiplo de 400. Por ejemplo 1900 no fue bisiesto, pero el año 2000 si lo será.

## Las Semanas

<http://www.phys.uu.nl/~vgent/calendar/isocalendar.htm>

<http://translate.google.com/translate?hl=es&langpair=en%7Ces&u=http://www.phys.uu.nl/~vgent/calendar/isocalendar.htm&rurl=translate.google.es>

### Numeración de las semanas

La numeración de las semanas de calendario se define la norma ISO (sección 2.2.10):

. [...] La primera semana del calendario de un año es la que incluye el primer jueves de ese año, y [...] la última semana del calendario de un año natural es la semana inmediatamente anterior a la primera semana del calendario de la próxima agenda año.

Otra forma de expresar la regla anterior es que la primera semana del año ISO es la primera semana que contiene al menos cuatro días del mes de enero. Del mismo modo, la última semana del año ISO es la última semana que contiene al menos cuatro días del mes de diciembre.

Como el jueves es el "día del medio" de la semana ISO, cada jueves, de un año calendario gregoriano pertenece al mismo período del año calendario ISO. Así, si un año gregoriano contiene 52 jueves el correspondiente año ISO tiene 52 semanas y si el año gregoriano contiene 53 jueves el correspondiente año ISO tiene 53 semanas.

Los números de semana ISO posibles están dentro del rango de 01 a 53. Un año tiene siempre 52 semanas y algunos años tienen 53. (Hay una excepción histórica: el año en que se introdujo el calendario gregoriano hubo menos de 365 días y menos de 52 semanas.)

### ¿Puede haber años con menos de una semana?

Por definición, la primera semana de un año es W01, es decir la semana 01, y por definición no puede haber un numero de semana inferior (porque esos días pertenecen a semanas del año anterior)

### ¿Puede haber años con 53 o más semanas?

Para responder a esta pregunta vamos a estudiar el peor caso posible, por ejemplo el año 1976, que es bisiesto y el día uno de enero es jueves, es decir en la primera semana del año tiene el máximo número posible de días del año anterior, es decir, 3 días (lunes 29, martes 30 y miércoles 31 de diciembre)

El último domingo del año 1976, es el domingo de la semana W52 y corresponde al día del año 361 ((52 \* 7) -3 = 361) por lo que quedan aun 5 días de diciembre (366-361) que pertenecen (por definición) a la semana W53 del año 1976. Es decir, en el peor de los casos existe la semana 53 pero no la semana 54, lo que demuestra que es posible la existencia de la semana 53

### ¿Es posible que existan años con 51 semanas o menos?:

Como en el ejemplo anterior vamos estudiar el peor de los casos, por ejemplo el año 1999 cuyo día uno de enero es viernes, lo que garantiza que los tres primeros días del año corresponden (por definición de semana ISO) a la última semana del año anterior.

El día 31 de diciembre de 1999 es viernes, luego la última semana del año 1999 es la que va desde 1999-12-27 al 2000-01-02. Si calculamos cual es el último domingo de la semana 52, vemos que corresponde al día del año 367 ((52 \* 7)+3 = 367) es decir hay dos días en esa semana (367-365) que corresponden al año 2000 (el día 1 sábado y el día 2 domingo). Lo que implica que aun en el peor de los casos posibles un año tiene 52 semanas es decir no puede haber un año con menos de 52 semanas

## El Formato de Fecha Week Date

Para una explicación de este formato de calendario, véase: [http://en.wikipedia.org/wiki/ISO\_week\_date](http://translate.googleusercontent.com/translate_c?hl=es&langpair=en%7Ces&u=http://en.wikipedia.org/wiki/ISO_week_date&rurl=translate.google.es&usg=ALkJrhire3clGjXKJia7YtpTCyUq8KZepw)

La norma ISO 8601 la define como una cadena formada de la siguiente manera YYYYWwwD

Donde:

* YYYY: Corresponde al año representado con cuatro cifras
* W: Es un carácter separador e identificador de formato de fecha
* ww: Es el numero de semana del año cuyos valores pueden ir de la semana 01 a la semana 53, se representa siempre con dos cifras, (ejemplo, la semana uno tiene el numero 01)
* D: Es el día de la semana cuyo rango va de 1 (que corresponde al lunes) hasta el 7 (que corresponde al domingo)

## Formatos de presentación según la norma ISO

Nota: La fecha que se muestra en los ejemplos siguientes es la que corresponde al viernes 4 de abril de 1985 (1985-04-12)

Completa (un día específico) (Week Date)

Formato Básico YYYYWwwD Ejemplo 1985W155

Formato extendido YYYY-Www-D Ejemplo 1985-W15-5

Reducida (una semana específica) (Calendar Week)

Formato Básico YYYYWww Ejemplo 1985W15

Formato extendido YYYY-Www Ejemplo 1985-W15

Expandida (mostrando el año con 6 dígitos)

* Completa (un día específico) (Week Date)

Formato Básico (+-)YYYYYWwwD Ejemplo +001985W155

Formato extendido (+-)YYYYY-Www-D Ejemplo +001985-W15-5

* Reducida (una semana específica) (Calendar Week)

Formato Básico (+-)YYYYYWww Ejemplo +001985W15

Formato extendido (+-)YYYYY-Www Ejemplo +001985-W15

# Cálculos diversos

## Averiguar el número de semanas de un año

Para calcular el número de semanas del año se puede usar el siguiente algoritmo

Fuente: [http://www.phys.uu.nl/~vgent/calendar/isocalendar.htm]

Un año calendario ISO y es largo (contiene 53 semanas), cuando:

* f (y) modulo 28 < 5

Un año calendario ISO y es corto (con 52 semanas), cuando:

* f (y)modulo 28 > 4

Siendo:

* f (y)= 5 \* y + 12 - 4 \* [Floor (y / 100) - Floor (y / 400)] + g (y) + h (y)
* g(y) = Floor(( y –100)/400) – Floor(( y –102)/400)
* h(y) = Floor(( y –200)/400) – Floor(( y –199)/400)
  + Las funciones g (y) y h (y) son funciones correctoras cuya única finalidad es la reparación de fallos numéricos en la ecuación anterior, en los años múltiplos de 100, 200 y 400.
  + Floor es una función que devuelve el número entero más grande menor o igual que el número decimal especificado.
  + Ejemplo: [http://msdn.microsoft.com/es-es/library/k7ycc7xh.aspx]
    - ' Value Floor
    - ' 7.03 7
    - ' 7.64 7
    - ' 0.12 0
    - ' -0.12 -1
    - ' -7.1 -8
    - ' -7.6 -8

**Otra forma de calcularlo es contando las semanas de un año**

Un año ISO tiene 52 o 53 semanas completas (364 o 371 días)

Referencia

* http://en.wikipedia.org/wiki/ISO\_week\_date
* http://translate.google.es/translate?hl=es&langpair=en%7Ces&u=http://en.wikipedia.org/wiki/ISO\_week\_date

Para calcular el número de semanas del año lo que haremos será contar las semanas y para ello vamos a incluir / quitar en los días del año los días que corresponden a semanas del año anterior/ siguiente.

Ejemplo, si el día uno de enero cae en jueves, esa semana es del año luego a efectos de cómputos de días el año tendrá tres días más (el lunes 29, el martes 30 y el miércoles 31 de diciembre del año pasado)

El mismo proceso se hace con la fecha del 31 de diciembre, al final tendré un número exacto de semanas o lo que es lo mismo un número de días que será o bien 364=52 semanas o bien 371=53 semanas. Divido esa cantidad por 7 y tendré el número de semanas que tiene ese año

Un ejemplo Numérico:

Contar el número de semanas del año 2009

El año 2009, el día 1 de enero cae en jueves y el día 31 de diciembre también cae en jueves

'--------------------------------------

' Jueves

' Dia uno de enero de 2009

' 29 30 31 1 2 3 4 Corrección = +3

'--------------------------------------

' Jueves

' Dia Treinta y uno de enero de 2009

' 28 29 30 31 1 2 3 Corrección = +3

'--------------------------------------

La semana que contiene del día 1 de enero de 2010, es una semana ISO que pertenece al año 2009 luego en realidad, a efectos de contar semanas del año tengo tres días más (el día 29, 30, y 31) , por eso sumare 3 días a los días del año

El día 31 es jueves y por lo tanto esa semana pertenece al año 2009, a la hora de contar por semanas tendré tres días más (el día 1, 2, y 3 del año 2010) por eso sumare 3 días al número de días del año

Días = 365 + (+3 + 3) = 371

371/7 = 53 semanas

Un ejemplo Numérico:

Contar el número de semanas del año 2010

El año 2010, el día 1 de enero cae en viernes y el día 31 de diciembre cae en (también) en viernes

'--------------------------------------

' viernes

' Dia uno de enero de 2010

' 28 29 30 31 1 2 3 Corrección = -3

'--------------------------------------

' viernes

' Dia Treinta y uno de enero de 2010

' 27 28 29 30 31 1 2 Corrección = +2

'--------------------------------------

La semana que contiene del día 1 de enero de 2010, es una semana ISO que pertenece al año anterior (2009) luego en realidad, a efectos de contar semanas del año tengo tres días menos (el día 1,2,y 3) que pertenecen a la última semana del año 2009, por eso restare 3 días a los días del año

El día 31 es jueves y por lo tanto esa semana pertenece al año 2010, a la hora de contar por semanas tendré dos días más (el día 1 y 2 del año 2011) por eso sumare 2 días al número de días del año

Días = 365 + (-3 +2) = 364

364/7 = 52 semanas

## Averiguar el número de semanas de un mes

El razonamiento es el mismo que hemos visto anteriormente, la diferencia es que un mes solo tiene o 4 o 5 semanas, ni más ni menos

Para que un mes tenga cinco semanas es necesario que el día uno sea lunes martes miércoles o jueves, es decir que el día uno este en la primera semana del mes, y también es necesario que el ultimo día del mes sea jueves Viernes, sábado o domingo, es decir que la última semana del mes pertenezca también al mes., dicho de otro modo, el mes tiene que tener cinco (5) jueves.

'------------------------------

' Ejemplo de mes con cinco(5) semanas

' Abril de 2010

' L M X J V S D

'------------------------------

' 29 30 31 1 2 3 4

' 5 6 7 8 9 10 11

' 12 13 14 15 16 17 18

' 19 20 21 22 23 24 25

' 26 27 28 29 30 1 2

Este algoritmo tiene un problema que es en los meses de febrero de los años normales en los que el día 1 es lunes, porque entonces, el mes es ‘cuadrado’ y cumple las condiciones descritas anteriormente, pero sin embargo el mes solo tiene cuatro semanas. Por este motivo este caso debe filtrarse previamente antes de aplicar el algoritmo descrito

'------------------------------

' Ejemplo del mes de febrero 'cuadrado'

' Febrero de 2010

' L M X J V S D

'------------------------------

' 1 2 3 4 5 6 7

' 8 9 10 11 12 13 14

' 15 16 17 18 19 20 21

' 22 23 24 25 26 27 28

'------------------------------

### Código VB.NET

Public Shared Function ISO8601SemanasDelMes( \_

ByVal año As Integer, \_

ByVal mes As Integer) As Integer

' ------------------------------------

Dim diasDelMes As Integer = Date.DaysInMonth(año, mes)

Dim diaUnoDelMes As Date = New Date(año, mes, 1)

Dim ultimoDiaDelMes As Date = New Date(año, mes, diasMes)

' ------------------------------------

' Todos los meses tienen siempre 4 semana

Dim numeroSemanasMes As Integer = 4

'---------------------------------------

' ver si tiene 5 semanas

'---------------------------------------

' problema con febrero, existen meses de febrero

' \*cuadrados\* el día 1 es lunes y el 28 domingo,

' cuatro (4) semanas justas

' pero en la prueba siguiente dan 5 semanas,

' por eso los elimino de la prueba

If diasDelMes > 28 Then

'---------------------------------------

' Si el dia uno de mes es

' lunes, o martes, o miercoles o jueves

'---------------------------------------

If diaUnoDelMes.DayOfWeek = DayOfWeek.Monday OrElse \_

diaUnoDelMes.DayOfWeek = DayOfWeek.Tuesday OrElse \_

diaUnoDelMes.DayOfWeek = DayOfWeek.Wednesday OrElse \_

diaUnoDelMes.DayOfWeek = DayOfWeek.Thursday Then

'---------------------------------------

' Y ADEMAS el dia fin de mes es

' jueves, o viernes, o sabado o domingo

'---------------------------------------

If ultimoDiaDelMes.DayOfWeek = DayOfWeek.Thursday OrElse \_

ultimoDiaDelMes.DayOfWeek = DayOfWeek.Friday OrElse \_

ultimoDiaDelMes.DayOfWeek = DayOfWeek.Saturday OrElse \_

ultimoDiaDelMes.DayOfWeek = DayOfWeek.Sunday Then

'---------------------------------------

' ENTONCES este mes tiene 5 semanas

numeroSemanasMes = 5

'---------------------------------------

End If ' / ultimoDiaDelMes

End If ' / diaUnoDelMes

End If ' / diasMes

'---------------------------------------

Return numeroSemanasMes

End Function

## Transformar una fecha Gregoriana a Ordinal

El formato de fecha que recibimos es (YYYY-MM-DD) y el que obtenemos es (YYYY-DDD)

El procedimiento es muy sencillo

* Calculamos el número de días que hay desde el día uno de enero hasta el día uno del mes de la fecha
* A continuación sumamos los días del mes

Para calcular los días acumulados se puede hacer de varias formas pero uno de los más sencillos es utilizar una tabla con los valores acumulados como la siguiente:

Cuadro con los días del año acumulados por meses

Dias acumulados: Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

Años normales : 0 31 59 90 120 151 181 212 243 273 304 334

Años bisiestos : 0 31 60 91 121 152 182 213 244 274 305 335

Un ejemplo numérico

Ejemplo: Trasformar una fecha Gregoriana a fecha Ordinal

2008-W39-5 que corresponde al día 2008-09-26,

y su fecha ordinal es 2008-270

Un ejemplo

Ejemplo: Viernes, 26 de septiembre 2008

(Ojo es bisiesto)

Día ordinal: 244(días hasta el 1 de septiembre) + 26 = 270

La fecha Week es 2008-270

## Transformar una fecha Gregoriana a fecha Week Date

El formato de fecha que recibimos es (YYYY-MM-DD) y el que obtenemos es (YYYY-Www-D)

El cálculo del año es directo, es el mismo

Para el día de la semana hay que usar la numeración de días de la semana ISO (entre el 1 de lunes a 7 para el domingo). Hay que calcular el día de la semana que es esa fecha. Para ello podemos utilizar las funciones de .NET (DayOfWeek) que proporcionan un resultado inmediato o bien utilizar el ***algoritmo de Zeller***. El resultado de cualquiera de estos métodos es un valor cuyo rango está entre el 0 =Domingo, hasta el 6= sábado. Pero la norma ISO 8601 indica que los días de la semana tienen el rango de 1= lunes hasta 7=domingo) por lo que habrá que realizar la oportuna transformación de los números de los días según el método que empleemos y el resultado que proporcione.

---------------------------------------------------------------------

Biblioteca de clases de .NET Framework

http://msdn.microsoft.com/es-es/library/ms229335.aspx

* DayOfWeek (Enumeración)
* <http://msdn.microsoft.com/es-es/library/system.dayofweek.aspx>
* DateTime.DayOfWeek (Propiedad)
* http://msdn.microsoft.com/es-es/library/system.datetime.dayofweek.aspx

Algoritmo de Zeller

* <http://www.matesymas.es/index.php?option=com_content&view=article&id=501:algoritmo-de-zeller&catid=179&Itemid=312>
* <http://es.wikipedia.org/wiki/Congruencia_de_Zeller>
* <http://latecladeescape.com/w0/con-nombre-propio/congruencia-de-zeller.-un-ingenioso-algoritmo-para-obtener-un-calendario-perpetuo.html>

---------------------------------------------------------------------

El cálculo del numero de semana es un poco más complicado, pero se puede calcular el número de semana de cualquier fecha a partir de su fecha ordinal (es decir, la posición dentro del año (YYYY-DDD)

El proceso es el siguiente:

* Usar los números de días de la semana ISO (entre el 1 de lunes a 7 para el domingo),
* Calcular el día Ordinal de la fecha
* Restar el día de la semana de la fecha de ordinal, a continuación, añadir 10.
* Dividir el resultado por 7.
* No haga caso de el resto, el cociente es igual al número de la semana.
* Correcciones :
  + Si el número de la semana así obtenido sea igual a 0, significa que la fecha dada pertenece a la última semana del año anterior.
    - Hay que realizar una llamada recursiva sobre el procedimiento para calcular el número de semana. Como la fecha pertenece a la última semana del año anterior, el cálculo de la semana hay que volverlo a realizar otra vez sobre el día 31 de diciembre del año anterior para obtener el año y el numero de semana correcto
  + Si se obtiene un número de semana igual a 53, la fecha puede ser (o no) de la semana siguiente (que es la primera semana (W01) del año siguiente) por lo que se debe comprobar este extremo.
    - La forma de comprobarlo es: Si la semana es la 53, y el mes 12 y el día mayor de 28 y es lunes, martes o miércoles, entonces la fecha es del año siguiente. En este caso no hay que volver a realizar los cálculos porque el resultado buscado es la primera semana del año ( semana W01) y el año es un año más del recibido en la fecha a estudiar

Cuadro con los días del año acumulados por meses

Dias acumulados: Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

Años normales : 0 31 59 90 120 151 181 212 243 273 304 334

Años bisiestos : 0 31 60 91 121 152 182 213 244 274 305 335

Un ejemplo numérico

Ejemplo: Trasformar una fecha Gregoriana a fecha Week Date

2008-W39-5 que corresponde al día 2008-09-26,

y su fecha ordinal es 2008-270

Un ejemplo

Ejemplo: Viernes, 26 de septiembre 2008

(Ojo es bisiesto)

Día ordinal: 244(días hasta el 1 de septiembre) + 26 = 270

Día de la semana: viernes = 5

270 - 5 + 10 = 275

275 / 7 = 39, más una fracción irrelevante

Resultados: Semana 39

La fecha Week es 2008-W39-5

Referencia Bibliográfica

<http://translate.google.es/translate?hl=es&langpair=en%7Ces&u>=  
http://en.wikipedia.org/wiki/ISO\_week\_date

## Transformar una fecha Week Date a fecha Ordinal

El formato de fecha que recibimos es (YYYY-Www-D) y el que obtenemos es (YYYY-DDD)

Hay varios procedimientos posibles, pero el más sencillo es el siguiente

* Sea [y] el año de la fecha ordinal; El cálculo del año es directo en principio es el mismo
* Calcular que día de la semana (lunes, martes, …) es el día cuatro de enero del año de la fecha (Usando, por ejemplo, el algoritmo de Zeller)
* Usar los números de días de la semana ISO (entre el 1 de lunes a 7 para el domingo). Corregir si es necesario el número del día de la semana
* Sumar tres al número obtenido y este será el valor de la corrección para las fechas de ese año
* Multiplica el número de la semana por 7, a continuación, agregar el día de la semana.
* Resta la corrección para el año
* Calcular los días del año de la fecha estudiada
* Sea {do} el resultado de la operación; que es el día del año de la fecha estudiada
  + Si el día ordinal así obtenido [do] es un cero o negativo, la fecha corresponde al año natural anterior.
    - El año de resultado se calcula restando una unidad al año de la fecha estudiada [y=y-1]
    - Calcular los días del nuevo año
    - El día del año se calcula restando los días del año obtenido menos el resultado negativo obtenido
  + Si el día ordinal así obtenido [do] es mayor que el número de días del año, la fecha corresponde al siguiente año natural.
    - El año es el mismo que el de la fecha estudiada
    - El día del año se vuelve a calcular restando los días del año obtenidos de los días del año estudiado

Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

2010-W04-7 que corresponde al día 2010-01-31,

y su fecha ordinal es 2010-031

Año = 2010

semana = 4

diaSemana = 7 [2010-W04-7]

diaSemana = (4-Enero-2010) = 1

Correccion = 1 + 3 = 4

diasOrdinal = (4 \* 7) + 7 = 35

diasOrdinal = 35 - 4 = 31

Fecha Ordinal calculada = 2010-031

Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

2011-W28-5 que corresponde al día 2011-07-15,

y su fecha ordinal es 2011-196

Año = 2011

semana = 28

diaSemana [2011-W28-5] = 5

diaSemana (4-Enero-2011) = 2

Correccion = 2 + 3 = 5

diasOrdinal = (28 \* 7) + 5 = 201

diasOrdinal = 201 - 5 = 196

Fecha Ordinal calculada = 2011-196

Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

2009-W53-5 que corresponde al día 2010-01-01,

y su fecha ordinal es 2010-001

Año = 2009

semana = 53

diaSemana [2009-W53-5] = 5

diaSemana (4-Enero-2009) = 7

Correccion = 7 + 3 = 10

diasOrdinal = (53 \* 7) + 5 = 376

diasOrdinal = 376 - 10 = 366

----- Correcciones -----

\*\* dias del año (2009) = 365

\*\* diasOrdinal = 366 - 365 = 1

\*\* Año (2009+1) = 2010

Fecha Ordinal calculada = 2010-001

Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

2009-W01-2 que corresponde al día 2008-12-30,

y su fecha ordinal es 2008-365

Año 2009

semana 1

diaSemana [2009-W01-2] = 2

diaSemana (4-Enero-2009) = 7

Correccion = 7 + 3 = 10

diasOrdinal = (1 \* 7) + 2 = 9

diasOrdinal = 9 - 10 = -1

-----Correcciones -----

\*\*\* Año (2009-1) = 2008

\*\*\* dias del año(2008) = 366 (nuevo calculo)

\*\*\* diasOrdinal = 366 + (-1) = 365

Fecha Ordinal calculada = 2008-365

## Transformar una fecha Week Date a fecha Gregoriana

La forma más fácil de hacerlo es transformar la fecha Calendario Semanas a fecha ordinal, y a continuación pasar la fecha ordinal a fecha gregoriana

* Ver el apartado Donde se describe el algoritmo para trasformar una fecha Week Date a Fecha Ordinal
* Ver el apartado Donde se describe el algoritmo para trasformar una fecha Ordinal a Fecha Gregoriana

## Transformar una fecha Ordinal a fecha Gregoriana

El formato de fecha que recibimos es (YYYY-DDD) y el que obtenemos es (YYYY-MM-DD)

La forma más fácil de hacerlo es transformar la fecha Calendario Semanas a fecha ordinal, siguiendo el procedimiento descrito anteriormente. El paso de fecha Ordinal a fecha gregoriana es muy fácil

Cuadro con los días del año acumulados por meses

Dias acumulados: Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

Años normales : 0 31 59 90 120 151 181 212 243 273 304 334

Años bisiestos : 0 31 60 91 121 152 182 213 244 274 305 335

Unos ejemplos

Ejemplo: Trasformar una fecha Week Date a Gregoriana

2009-W01-2 que corresponde al día 2008-12-30,

y su fecha ordinal es 2008-365

Ejemplo el día: 2008-365 (Ojo es bisiesto)

Días acumulados = 335 (días hasta el 1 de diciembre)

365 – 335 = 30 (El día 30 de Diciembre)

El resultado es el 30 de diciembre de 2008 (2008-12-30)

Ejemplo: Trasformar una fecha Week Date a Gregoriana

2011-W28-5 que corresponde al día 2011-07-15,

y su fecha ordinal es 2011-196

Ejemplo el día: 2011-196

Días acumulados =181 (días hasta el 1 de julio)

196 – 181 = 15 (El día 15 de Julio)

El resultado es el 15 de Julio de 2011 (2011-07-15)

## Transformar una fecha Ordinal a fecha Week Date

El formato de fecha que recibimos es (YYYY-DDD) y el que obtenemos es (YYYY-Www-D)

* Ver el apartado Donde se describe el algoritmo para trasformar una fecha Gregoriana a Fecha Week Date. Allí se describe como se calcula primer la fecha ordinal y a continuación se hacen los cálculos para obtener la fecha Week Date

# Código VB.NET

## Averiguar el número de semanas de un mes

''' <summary>

''' Calcula el número de semanas que tiene un mes

''' </summary>

''' <param name="año">El año</param>

''' <param name="mes">El Mes del que se cuentan el

''' número de semanas</param>

''' <returns>El número de semanas del mes, un valor

''' entero dentro del rango 4-5</returns>

Public Overloads Shared Function ContarSemanasDelMes2( \_

ByVal año As Integer, ByVal mes As Integer) As Integer

Dim diasMes As Integer = Date.DaysInMonth(año, mes)

Dim unoMes As Date = New Date(año, mes, 1)

Dim FinDeMes As Date = New Date(año, mes, diasMes)

' valor estandar

Dim semanasMes As Integer = 4

'------------------------------

' Abril de 2010 con cinco(5) semanas

' L M X J V S D

'------------------------------

' 29 30 31 1 2 3 4

' 5 6 7 8 9 10 11

' 12 13 14 15 16 17 18

' 19 20 21 22 23 24 25

' 26 27 28 29 30 1 2

'---------------------------------------

' ver si tiene 5 semanas

'---------------------------------------

' problema con febrero, existen meses de febrero 'cuadrados'

' el día 1 es lunes y el 28 domingo, cuatro semanas justas

' pero en la prueba siguiente dan 5 semanas, por eso

' los elimino de la prueba

'------------------------------

' Febrero de 2010

' L M X J V S D

'------------------------------

' 1 2 3 4 5 6 7

' 8 9 10 11 12 13 14

' 15 16 17 18 19 20 21

' 22 23 24 25 26 27 28

'------------------------------

If diasMes > 28 Then

'------------------------------

' Abril de 2010 con cinco(5) semanas

' L M X J V S D

'------------------------------

' 29 30 31 1 2 3 4

' 5 6 7 8 9 10 11

' 12 13 14 15 16 17 18

' 19 20 21 22 23 24 25

' 26 27 28 29 30 1 2

' el dia uno de mes debe ser lunes, martes,

' miercoles o jueves

If unoMes.DayOfWeek = System.DayOfWeek.Monday OrElse \_

unoMes.DayOfWeek = System.DayOfWeek.Tuesday OrElse \_

unoMes.DayOfWeek = System.DayOfWeek.Wednesday OrElse \_

unoMes.DayOfWeek = System.DayOfWeek.Thursday Then

' Y ADEMAS el dia fin de mes tiene que ser

' jueves, viernes, sabado o domingo

If FinDeMes.DayOfWeek = System.DayOfWeek.Thursday OrElse \_

FinDeMes.DayOfWeek = System.DayOfWeek.Friday OrElse \_

FinDeMes.DayOfWeek = System.DayOfWeek.Saturday OrElse \_

FinDeMes.DayOfWeek = System.DayOfWeek.Sunday Then

'-------------

' este mes tiene 5 semanas

semanasMes = 5

End If

End If

End If ' / diasMes > 28

Return semanasMes

End Function

## Averiguar el número de semanas de un año

''' <summary>

''' Devuelve las semanas del año

''' </summary>

''' <param name="unAño">el año cuyas semans queremos conocer</param>

''' <returns>un numero integer que representa el numero de

''' semanas del año</returns>

''' <remarks>

''' <code>http://www.phys.uu.nl/~vgent/calendar/isocalendar.htm</code>

''' </remarks>

Public Overloads Shared Function ContarSemanasDelAño( \_

ByVal unAño As Integer) As Integer

'---------------------------------------------------------------

' Un año calendario ISO y es largo (contiene 53 semanas), cuando:

'f( y ) modulo 28 < 5

' Un año calendario ISO y es corto (con 52 semanas), cuando:

'f( y ) modulo 28 > 4

' con:

' f (y) = 5 \* y + 12 - 4 \*

' [Floor (y / 100) - Floor (y / 400)] + g (y) + h (y)

' y:

' g( y ) = floor(( y –100)/400) – floor(( y –102)/400)

' h( y ) = floor(( y –200)/400) – floor(( y –199)/400)

' Las funciones g (y) y h (y) son "factores correctores"

' cuya única finalidad es la reparación de fallos

' numéricos en la ecuación anterior, cerca del

' año 100 + 400 + 200 K y 400 K.

'---------------------------------------------------------------

Dim h As Integer = 0

Dim y As Integer = unAño

'h( y ) = floor(( y –200)/400) – floor(( y –199)/400)

h = CType(Math.Floor((y - 200) / 400), Integer) - \_

CType(Math.Floor((y - 199) / 400), Integer)

'g( y ) = floor(( y –100)/400) – floor(( y –102)/400)

Dim g As Integer = 0

g = CType(Math.Floor((y - 100) / 400), Integer) - \_

CType(Math.Floor((y - 102) / 400), Integer)

' f (y) = 5 \* y + 12 - 4 \*[Floor (y / 100) - Floor (y / 400)] \_

+ g (y) + h (y)

Dim f As Integer = 0

f = CType(5 \* y + 12 - 4 \* \_

(Math.Floor(y / 100) - Math.Floor(y / 400)) + g + h, \_

Integer)

If (f Mod 28) < 5 Then Return 53

If (f Mod 28) > 4 Then Return 52

End Function

## Calculo del número de la semana de una fecha

''' <summary>

''' Calcula la semana de una fecha segun la convencion ISO 8601

''' </summary>

''' <param name="unaFecha">

''' La fecha cuya semana se quiere calcular</param>

''' <returns>

''' Un objeto [System.Collections.Generic. \_

''' KeyValuePair(Of Integer, Integer)]

''' cuya clave (Key) contiene el año, y

''' la propiedad Value contiene la semana

'''</returns>

''' <remarks>

''' La primera semana del año es la que contiene

''' el primer jueves del año

''' </remarks>

Public Shared Function CalculoSemana( \_

ByVal unaFecha As DateTimeOffset) \_

As System.Collections.Generic.KeyValuePair(Of Integer, Integer)

'------------------------------------------

' Referencia Bibliográfica

' <http://translate.google.es/translate?hl=es&langpair=en%7Ces&u>=

' http://en.wikipedia.org/wiki/ISO\_week\_date

' Se puede calcular el número de semana de cualquier fecha a

' partir de su fecha ordinal (es decir, la posición dentro

' del año)y su día de la semana.

'------------------------------------------

' Proceso

' Calcular el dia Ordinal de la fecha

' Usar los números de días de la semana ISO

' (entre el 1 de lunes a 7 para el domingo),

' Restar el día de la semana de la fecha de ordinal,

' a continuación, añadir 10.

' Dividir el resultado por 7.

' No haga caso de el resto, el cociente es igual al

' número de la semana.

' Correcciones :

' - Si el número de la semana así obtenido sea

' igual a 0, significa que la fecha dada

' pertenece a la ultima semana del año anterior.

' - Si se obtiene un número de semana del 53,

' se debe comprobar que la fecha no es

' la semana 1 del año siguiente.

' -----------------------------------------------

' Ejemplo: Viernes, 26 de septiembre 2008

' Día ordinal: 244( dias hasta sepiembre) + 26 = 270

' Día de la semana: viernes = 5

' 270 - 5 + 10 = 275

' 275 / 7 = 39, más una fracción irrelevante

' Resultados: Semana 39

' -----------------------------------------------

' -----------------------------------------------

' Calculos

' Para el devolver el resultado

' contiene Key= año; Value = semana de la fecha

Dim parejaAñoSemana As System.Collections.Generic. \_

KeyValuePair(Of Integer, Integer)

'--------------------------------------

' Valores para realizar los calculos

Dim salidaAño As Integer = unaFecha.Year

Dim salidaNumeroSemana As Integer = 0

' OJO el dia de la semana es ISO 8601

' Coincide con la enumeracion [DayOfWeek] excepto el domingo

Dim SalidaDiaDeLaSemana As Integer = unaFecha.DayOfWeek

If unaFecha.DayOfWeek = System.DayOfWeek.Sunday Then

SalidaDiaDeLaSemana = 7

End If

'--------------------------------------

' calculo del dia ORDINAL del año correspondiente a la fecha

Dim diaOrdinal As Integer = unaFecha.DayOfYear

'--------------------------------------

' calculos del numero de la semana

Dim Dividendo As Integer = ((diaOrdinal + 10) - SalidaDiaDeLaSemana)

Dim restoDivision As Integer = 0 ' no se usa para nada

'Cociente Division = Math.DivRem(Dividendo, Divisor, restoDivision)

salidaNumeroSemana = Math.DivRem(Dividendo, 7, restoDivision)

'-------------------------------------------------

' Comprobacion de resultado salidaNumeroSemana = 0

If salidaNumeroSemana = 0 Then

'---------------------------------------

' !!! ATENCION !!!

' la semana corresponde al año anterior

'---------------------------------------

' LLAMADA RECURSIVA

Dim nuevafecha As New Date( \_

unaFecha.Year - 1, 12, 31, 0, 0, 0, DateTimeKind.Unspecified)

Dim parejaAux As New System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)(salidaAño, salidaNumeroSemana)

' hacer la llamada recursiva

parejaAux = CalculoSemana(nuevafecha)

' recoger el resultado

salidaAño = parejaAux.Key

salidaNumeroSemana = parejaAux.Value

End If ' / salidaNumeroSemana = 0

'-------------------------------------------------

' Comprobacion de resultado salidaNumeroSemana = 53

If salidaNumeroSemana = 53 Then

'---------------------------------------

' !!! CUIDADO !!!

' la semana PUEDE corresponde al año siguiente (o no)

'---------------------------------------

' Si el día 31 del año es lunes, martes o miércoles

' según la norma ISO, esa semana pertenece al año siguiente

' es decir si es el 29 lunes , o 30 martes ,

' o 31 miércoles de diciembre

' la semana es la 1 del año siguiente

' L M X J V S D

' 29 30 31 1 2 3 4

If unaFecha.Month = 12 Then

If unaFecha.Day > 28 Then

If unaFecha.Day = 29 AndAlso unaFecha.DayOfWeek < 2 OrElse \_

unaFecha.Day = 30 AndAlso unaFecha.DayOfWeek < 3 OrElse \_

unaFecha.Day = 31 AndAlso unaFecha.DayOfWeek < 4 Then

salidaAño = unaFecha.Year + 1

salidaNumeroSemana = 1

End If

End If

End If

End If ' / salidaNumeroSemana = 53

parejaAñoSemana = New System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)(salidaAño, salidaNumeroSemana)

Return parejaAñoSemana

End Function

## Transformar una fecha Gregoriana a fecha Ordinal

''' <summary>

''' Convierte una fecha Gregoriana en una fecha ordinal

''' (YYYY-MM-DD) -->> (YYYY-DDD)

''' </summary>

''' <param name="año">El año entre 1 y 9999</param>

''' <param name="mes">El mes entre 1 y 12</param>

''' <param name="dia">El dia entre 1 y 31</param>

''' <returns>

''' Un objeto [System.Collections.Generic.KeyValuePair( \_

''' Of Integer, Integer)] que contiene en el

''' campo Key= año y en su campo Value = dia ordinal del año

'''</returns>

Public Shared Function ConvertGregorianaToOrdinal( \_

ByVal año As Integer, \_

ByVal mes As Integer, \_

ByVal dia As Integer) As \_

System.Collections.Generic.KeyValuePair(Of Integer, Integer)

'--------------------------------------------------------------

'El procedimiento es muy sencillo

'• Calculamos el número de días que hay desde el día

' uno de enero hasta el día uno del mes de la fecha

'• A continuación sumamos los días del mes

'

' Para calcular los días acumulados se puede hacer de varias

' formas pero uno de los más sencillos es utilizar una

' tabla con los valores acumulados como la siguiente:

'------------------------------------------------------------

'Cuadro con los días del año acumulados por meses

'Dias acumulados: Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

'Años normales : 0 31 59 90 120 151 181 212 243 273 304 334

'Años bisiestos : 0 31 60 91 121 152 182 213 244 274 305 335

'----------------------------------------------------------------

'Un ejemplo numérico

'Ejemplo: Trasformar una fecha Gregoriana a fecha Ordinal

'2008-W39-5 que corresponde al día 2008-09-26,

' y su fecha ordinal es 2008-270

' Un(ejemplo)

'Ejemplo: Viernes, 26 de septiembre 2008

' (Ojo es bisiesto)

'Día ordinal: 244(días hasta el 1 de septiembre) + 26 = 270

'La fecha Week es 2008-270

'----------------------------------------------------------------

'----------------------------------------------

'Control de parametros

'----------------------------------------------

If año < 1 OrElse año > 9999 Then

Throw New ArgumentOutOfRangeException( \_

"Año", año, "Año (del 1 al 9999)")

End If

'----------------------------------------------

If mes < 1 OrElse mes > 12 Then

Throw New ArgumentOutOfRangeException( \_

"mes", mes, "mes (del 1 al 12)")

End If

'----------------------------------------------

'Calculos previos al control del Dia del mes

'----------------------------------------------

Dim EsBisiesto As Boolean

EsBisiesto = (año Mod 4 = 0 And \_

(año Mod 100 <> 0 Or año Mod 400 = 0))

Dim diasQueTieneElMes As Integer = 0

Select Case mes

Case 2

If EsBisiesto Then

diasQueTieneElMes = 29

Else

diasQueTieneElMes = 28

End If

Case 4, 6, 9, 11

diasQueTieneElMes = 30

Case 1, 3, 5, 7, 8, 10, 12

diasQueTieneElMes = 31

End Select

'----------------------------------------------

If dia < 1 OrElse dia > diasQueTieneElMes Then

Throw New ArgumentOutOfRangeException( \_

"dia", dia, "dia (del 1 al " & diasQueTieneElMes & ")")

End If

'----------------------------------------------

'----------------------------------------------

' Matriz de dias acumulados

'----------------------------------------------

Dim diasAcumulados() As Integer

OJO, indice de la matriz en base 0 = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Dim diasAcumuladosNormales() As Integer = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334}

Dim diasAcumuladosBisiesto() As Integer = {0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335}

If EsBisiesto = True Then

' es bisiesto

diasAcumulados = diasAcumuladosBisiesto

Else

' es normal

diasAcumulados = diasAcumuladosNormales

End If

'----------------------------------------------

' Calculos

'----------------------------------------------

Try

Dim auxAño As Integer = año

Dim auxDiasOrdinal As Integer = 0

'--------------------------------------------------------

'Ejemplo: Viernes, 26 de septiembre 2008

' (Ojo es bisiesto)

'Día ordinal: 244(días hasta el 1 de septiembre) + 26 = 270

'La fecha Week es 2008-270

'--------------------------------------------------------

' Para recuperar los dias acumulados tendre

' que usar mes-1 porque la matriz esta en indice cero

auxDiasOrdinal = diasAcumulados(mes - 1) + dia

'------------------------------------------

' Devolver el valor obtenido

' Un objeto [System.Collections.Generic.KeyValuePair \_

' (Of Integer, Integer)] que contiene

' en el campo Key= año y en su

' campo Value = dia ordinal del año

Dim pareja As System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)

pareja = New System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)(año, auxDiasOrdinal)

Return pareja

Catch ex As Exception

Throw

End Try

End Function

## Transformar una fecha Ordinal a fecha Gregoriana

''' <summary>

''' Funcion que transforma una fecha ordinal en fecha gregoriana

''' </summary>

''' <param name="año">El año</param>

''' <param name="diaDelAño">El dia del año</param>

''' <returns></returns>

''' <remarks></remarks>

Public Shared Function ConvertOrdinalToGregoriana( \_

ByVal año As Integer, ByVal diaDelAño As Integer) As Date

'----------------------------------------------

'Control de parametros

'----------------------------------------------

'--------------------------------------------------------

If año < 1 OrElse año > 9999 Then

Throw New ArgumentOutOfRangeException( \_

"Año", año, "Año (del 1 al 9999)")

End If

'--------------------------------------------------------

Dim diasQueTieneElAño As Integer = 365

If (año Mod 4 = 0 And \_

(año Mod 100 <> 0 Or año Mod 400 = 0)) = True Then

diasQueTieneElAño = 366

End If

If diaDelAño > diasQueTieneElAño Then

Throw New ArgumentOutOfRangeException("diaDelAño", diaDelAño, \_ String.Format(System.Globalization.CultureInfo.CurrentCulture, \_

"El dia del año: {0} de la fecha ordinal recibida [{1}-{2:000}] " & \_

"supera al numero de dias que tiene ese año: {3}={4} dias ", \_

diaDelAño, año, diaDelAño, año, diasQueTieneElAño))

End If

'----------------------------------------------

' Matriz de dias acumulados

'----------------------------------------------

Dim diasAcumulados() As Integer

' OJO, indice de la matriz en base 0 = \_

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Dim diasAcumuladosNormales() As Integer = \_

{0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334}

Dim diasAcumuladosBisiesto() As Integer = \_

{0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335}

If diasQueTieneElAño = 366 Then

' es bisiesto

diasAcumulados = diasAcumuladosBisiesto

Else

' es normal

diasAcumulados = diasAcumuladosNormales

End If

'----------------------------------------------

' Calculos

'----------------------------------------------

Dim mes As Integer = 0

For mes = 0 To 11

If diasAcumulados(mes) > diaDelAño Then

'me he pasado

' Dias acumulados: Ene Feb Mar Abr May Jun Jul Ago Sep Oct Nov Dic

' Años normales : 0 31 59 90 120 151 181 212 243 273 304 334

' Ejemplo

' dia Del Año = 196

' Dias acumulados = 212

' el mes es Julio no agosto

' Al salir por [exit for], el mes suma uno y entonces

' queda dento del rango de 1 a 12

' Sin embargo para recuperar los dias acumulados tendre

' que usar mes-1 porque la matriz esta en indice cero

Exit For

End If

Next mes

Dim dia As Integer

dia = diaDelAño - diasAcumulados(mes - 1)

Return New Date(año, mes, dia)

End Function

## Transformar una fecha Ordinal a fecha Week Date

''' <summary>

''' Convierte una fecha Ordinal (YYYY\*DDD)

''' en una fecha WeekDate (YYYY-Www-D)

''' </summary>

''' <param name="año">El año</param>

''' <param name="diaOrdinalDelAño">

''' El dia ordinal delaño contando desde el uno de enero</param>

''' <returns>una cadena que contiene una fecha Week Date

''' con formato Extendido [YYYY-Www-D] </returns>

''' <remarks>

''' Utiliza la funcion [ConvertOrdinalToGregoriana]

''' Utiliza la funcion [CalculoSemana]

'''</remarks>

Public Shared Function ConvertOrdinalToWeek( \_

ByVal año As Integer, \_

ByVal diaOrdinalDelAño As Integer) As String

'----------------------------------------------

'Control de parametros

'----------------------------------------------

If año < 1 OrElse año > 9999 Then

Throw New ArgumentOutOfRangeException( \_

"Año", año, "Año (del 1 al 9999)")

End If

'--------------------------------------------------------

Dim diasQueTieneElAño As Integer = 365

If (año Mod 4 = 0 And \_

(año Mod 100 <> 0 Or año Mod 400 = 0)) = True Then

diasQueTieneElAño = 366

End If

If diaOrdinalDelAño > diasQueTieneElAño Then

Throw New ArgumentOutOfRangeException( \_

"diaDelAño", diaOrdinalDelAño, \_

String.Format(System.Globalization.CultureInfo.CurrentCulture, \_

"El dia del año: {0} de la fecha ordinal recibida [{1}-{2:000}] " & \_

"supera al numero de dias que tiene ese año: {3}={4} dias ", \_

diaOrdinalDelAño, año, diaOrdinalDelAño, año, diasQueTieneElAño))

End If

'--------------------------------------------------------

' -----------------------------------------------

' Apunte Táctico

' -----------------------------------------------

' Referencia Bibliográfica

' http://en.wikipedia.org/wiki/ISO\_week\_date

' Se puede calcular el número de semana de cualquier

' fecha a partir de su fecha ordinal (es decir, la

' posición dentro del año) y su día de la semana.

' -----------------------------------------------

' Proceso

' Usar una fecha Ordinal

' Primero calcular la fecha gregoriana

' (Necesito conocer el dia de la semana)

' Usar los números de días de la semana ISO

' (entre el 1 de lunes a 7 para el domingo),

' Restar el día de la semana de la fecha de ordinal,

' a continuación, añadir 10.

' Dividir el resultado por 7.

' No haga caso de el resto, el cociente es

' igual al número de la semana.

' Correcciones :

' - Si el número de la semana así obtenido sea

' igual a 0, significa que la fecha dada pertenece

' a la ultima semana del año anterior.

' - Si se obtiene un número de semana del 53, se debe

' comprobar que la fecha no es realmente en la

' semana 1 del año siguiente.

' -----------------------------------------------

' Ejemplo: Viernes, 26 de septiembre 2008

' Día ordinal: 244( dias hasta sepiembre) + 26 = 270

' Día de la semana: viernes = 5

' 270 - 5 + 10 = 275

' 275 / 7 = 39, más una fracción irrelevante

' Resultados: Semana 39

' -----------------------------------------------

' -----------------------------------------------

' CALCULOS

' -----------------------------------------------

' 1) Obtener la fecha gregoriana

' 2) Calcular el numero de la semana

' 3) Calcular el dia de la semana

' 4) Devoler el resultado en una cadena

' -----------------------------------------------

' Variables para recoger resultados parciales

Dim auxAño As Integer = 0

Dim auxSemana As Integer = 0

Dim auxDiaSemanaISO As Integer = 0

Dim auxFechaGregoriana As Date

Try

' -----------------------------------------------

' 1) Obtener la fecha gregoriana

auxFechaGregoriana = \_

ConvertOrdinalToGregoriana(año, diaOrdinalDelAño)

' -----------------------------------------------

' 2) Calcular el numero de la semana

' Utilizo una funcion auxiliar que se encarga de ese trabajo

' Devuelve el resultado como

' [Generic.KeyValuePair(Of Integer, Integer)]

' donde: Key= año; Value = semana de la fecha

Dim parejaAñoSemana As System.Collections.Generic. \_

KeyValuePair(Of Integer, Integer)

parejaAñoSemana = CalculoSemana(auxFechaGregoriana)

auxAño = parejaAñoSemana.Key

auxSemana = parejaAñoSemana.Value

' -----------------------------------------------

' 3) Calcular el dia de la semana

' OJO el dia de la semana es ISO 8601

' Coincide con la enumeracion [DayOfWeek] excepto el domingo

auxDiaSemanaISO = auxFechaGregoriana.DayOfWeek

If auxFechaGregoriana.DayOfWeek = \_

System.DayOfWeek.Sunday Then

auxDiaSemanaISO = 7

End If

' -----------------------------------------------

' 4) Devoler el resultado en una cadena

Dim provider As System.IFormatProvider = \_

Globalization.CultureInfo.CurrentCulture

Dim cadenaFormato As String = "{0:0000}-W{1:00}-{2:0}"

Return String.Format( \_

provider, cadenaFormato, \_

auxAño, auxSemana, auxDiaSemanaISO)

Catch ex As Exception

Throw

End Try

End Function

## Transformar una fecha Week Date a fecha Ordinal

''' <summary>

''' Trasforma una fecha en formato WeekDate

''' [yyyy-Www-d] en fecha Ordinal [yyyy-ddd]

''' </summary>

''' <param name="año">El año</param>

''' <param name="semana">la semana</param>

''' <param name="diaDeLaSemana">

''' El dia de la semana (1=lunes; 7=domingo)</param>

''' <returns>

''' Un objeto [System.Collections.Generic.KeyValuePair(Of Integer, Integer)]

''' que contiene en el campo contiene Key= año y

''' en su campo Value = dia ordinal del año

''' </returns>

''' <exception cref="ArgumentOutOfRangeException">

''' <para>o bien <paramref name="año">año</paramref> Año (del 1 al 9999). </para>

''' <para>o bien <paramref name="año">semana</paramref> semana (del 52 al 53). </para>

''' <para>o bien <paramref name="año">diaDeLaSemana</paramref>diaDeLaSemana (del 1=lunes al 7=domingo). </para>

'''</exception>

Public Overloads Shared Function ConvertWeekToOrdinal( \_

ByVal año As Integer, \_

ByVal semana As Integer, \_

ByVal diaDeLaSemana As Integer) As \_

System.Collections.Generic.KeyValuePair(Of Integer, Integer)

'----------------------------------------------

'Control de parametros

'--------------------------------------------------------

If año < 1 OrElse año > 9999 Then

Throw New ArgumentOutOfRangeException( \_

"Año", año, "Año (del 1 al 9999)")

End If

'--------------------------------------------------------

Dim semanasQueTieneElAño As Integer = 52

'FALTA EL CALULO DE SEMANAS DEL AÑO

'If semana <> semanasQueTieneElAño Then

' falta poner aqui la excepcion

'End If

If semana < 52 Or semana > 53 Then

Throw New ArgumentOutOfRangeException("semana", semana, \_

String.Format(System.Globalization.CultureInfo.CurrentCulture, \_

"la semana {0} de la fecha Week recibida [{1:0000}-W{2:00}-{3:0}] " & \_

"supera al numero de semanas que tiene ese año: {4:0000} = W{5:00} semanas ", \_

semana, año, semana, diaDeLaSemana, año, semanasQueTieneElAño))

End If

'----------------------------------------------

If diaDeLaSemana < 1 OrElse diaDeLaSemana > 7 Then

Throw New ArgumentOutOfRangeException( \_

"diaDeLaSemana", diaDeLaSemana, \_

"Dia de la semana (del 1=lunes al 7=domingo)")

End If

'--------------------------------------------------------

'---------------------------------------------------------------

' Calculos

'---------------------------------------------------------------

Try

'---------------------------------------------------------------

' - Sea [y] el año de la fecha ordinal; El cálculo del año es directo

' en principio es el mismo

' - Calcular que día de la semana (lunes, martes, …) es el día

' cuatro de enero del año de la fecha

' (Usando, por ejemplo, el algoritmo de Zeller)

' - Usar los números de días de la semana ISO (entre el 1 de lunes

' a 7 para el domingo). Corregir si es necesario el

' número del día de la semana

' - Sumar tres al número obtenido y este será el valor de la

' corrección para las fechas de ese año

' - Multiplica el número de la semana por 7, a continuación,

' agregar el día de la semana.

' - Resta la corrección para el año

' - Calcular los días del año de la fecha estudiada

' - Sea {RO} el resultado de la operación; que es el día

' del año de la fecha estudiada

' + Si el día ordinal así obtenido {RO} es un cero o negativo,

' la fecha corresponde al año natural anterior.

' - El año de resultado se calcula restando una unidad

' al año de la fecha estudiada [y=y-1]

' - Calcular los días del nuevo año

' - El día del año se calcula restando los días del año

' obtenido menos el resultado negativo obtenido

' + Si el día ordinal así obtenido {RO} es mayor que el

' número de días del año, la fecha corresponde al

' siguiente año natural.

' - El año es el mismo que el de la fecha estudiada

' - El día del año se vuelve a calcular restando los días

' del año obtenidos de los días del año estudiado

'---------------------------------------------------------------

'Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

'2010-W04-7 que corresponde al día 2010-01-31,

' y su fecha ordinal es 2010-031

'

' Año = 2010

' semana = 4

' diaSemana = 7 [2010-W04-7]

' diaSemana = (4-Enero-2010) = 1

'

' Correccion = 1 + 3 = 4

' diasOrdinal = (4 \* 7) + 7 = 35

' diasOrdinal = 35 - 4 = 31

'

' Fecha Ordinal calculada = 2010-031

'---------------------------------------------------------------

'Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

'2011-W28-5 que corresponde al día 2011-07-15,

' y su fecha ordinal es 2011-196

'

' Año = 2011

' semana = 28

' diaSemana [2011-W28-5] = 5

' diaSemana (4-Enero-2011) = 2

'

' Correccion = 2 + 3 = 5

' diasOrdinal = (28 \* 7) + 5 = 201

' diasOrdinal = 201 - 5 = 196

'

' Fecha Ordinal calculada = 2011-196

'---------------------------------------------------------------

'Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

'2009-W53-5 que corresponde al día 2010-01-01,

' y su fecha ordinal es 2010-001

'

' Año = 2009

' semana = 53

' diaSemana [2009-W53-5] = 5

' diaSemana (4-Enero-2009) = 7

'

' Correccion = 7 + 3 = 10

' diasOrdinal = (53 \* 7) + 5 = 376

' diasOrdinal = 376 - 10 = 366

' ----- Correcciones -----

' \*\* dias del año (2009) = 365

' \*\* diasOrdinal = 366 - 365 = 1

' \*\* Año (2009+1) = 2010

'Fecha Ordinal calculada = 2010-001

'---------------------------------------------------------------

'Ejemplo: Trasformar una fecha Week Date a Fecha Ordinal

'2009-W01-2 que corresponde al día 2008-12-30,

' y su fecha ordinal es 2008-365

'

' Año 2009

' semana 1

' diaSemana [2009-W01-2] = 2

' diaSemana (4-Enero-2009) = 7

'

' Correccion = 7 + 3 = 10

' diasOrdinal = (1 \* 7) + 2 = 9

' diasOrdinal = 9 - 10 = -1

'-----Correcciones -----

' \*\*\* Año (2009-1) = 2008

' \*\*\* dias del año(2008) = 366 (nuevo calculo)

' \*\*\* diasOrdinal = 366 + (-1) = 365

' Fecha Ordinal calculada = 2008-365

'---------------------------------------------------------------

'---------------------------------------------------------------

' dia de la semana que es el dia 4 de enero del año de la fecha

Dim DiaSemanaCuatroEnero As Integer = \_

ISO8601WeekDate.DiaDeLaSemanaISO( \_

New DateTimeOffset( \_

año, 1, 4, 0, 0, 0, System.TimeSpan.Zero))

' la correccion estandar

Dim correccion As Integer = DiaSemanaCuatroEnero + 3

Dim diasOrdinal As Integer = 0

diasOrdinal = (semana \* 7) + diaDeLaSemana

diasOrdinal = diasOrdinal - correccion

' si el dia ordinal es mayor que el numero de dias del año

' entonces es del año siguiente

Dim diasAñoActual As Integer = 365

If Date.IsLeapYear(año) = True Then diasAñoActual = 366

If diasOrdinal > diasAñoActual Then

diasOrdinal = diasOrdinal - diasAñoActual

año = año + 1

End If

' si el dia ordinal es cero o menor que cero es del año anterior

' hay que volver a clacular los dias dle año por si es bisiesto

If diasOrdinal < 1 Then

año = año - 1

Dim diasAñoAnterior As Integer = 365

If Date.IsLeapYear(año) = True Then diasAñoAnterior = 366

diasOrdinal = diasAñoAnterior + diasOrdinal

End If

'------------------------------------------

' Devolver el valor obtenido

' Un objeto [System.Collections.Generic.KeyValuePair( \_

' Of Integer, Integer)] que contiene

' en el campo Key= año y en su campo Value = dia ordinal del año

Dim pareja As System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)

pareja = New System.Collections.Generic.KeyValuePair( \_

Of Integer, Integer)(año, diasOrdinal)

Return pareja

Catch ex As Exception

Throw

End Try

End Function

## Transformar una fecha Gregoriana a fecha Week Date

La forma más fácil de hacerlo es:

* Transformar la fecha gregoriana en ordinal
* Transformar la fecha ordinal en fecha WeekDate

## Transformar una fecha Week Date a fecha Gregoriana

La forma más fácil de hacerlo es:

* Transformar la fecha WeekDate en fecha ordinal
* Transformar la fecha ordinal en gregoriana