

Artículo

[Kurro Lopez](#) · 11 feb, 2022 · Lectura de 5 min

Cómo convertirse en un señor del tiempo - El nacimiento

El tiempo dirá, siempre lo hace.

El Doctor.

No es una tarea fácil dominar fechas y horas, siempre es un problema y a veces confuso en cualquier lenguaje de programación, vamos a aclarar y a poner unos cuantos tips para que esta tarea sea lo más sencilla posible.

Súbete a la TARDIS que te voy a convertir en un Señor del tiempo



Empecemos por lo básico

Si vienes de otros lenguajes, comentar que las fechas en InterSystems Object Script (en adelante IOS, no confundir con los móviles) son un poco peculiares.

Cuando ejecutamos el comando [\\$HOROLOG](#) en el terminal, para tener la fecha y hora actual, verás que se divide en dos partes:

```
WRITE $HOROLOG
```

```
> 66149,67164
```

El primer valor es el día, para ser exactos el número de días desde el 31 de diciembre de 1840, osea, que el valor 1 es el 1 de enero de 1841 y el segundo los segundos desde la 00:00 de ese día.

En este ejemplo, 66149 corresponde con 09/02/2022 y 67164 con las 18:39:24 horas. A este formato, lo vamos a llamar formato interno de fecha y hora.

¿Confuso? pues vamos a empezar a desvelar los grandes secretos del universo (de fechas y horas)

¿Cómo puedo convertir el formato interno en formato mas claro?

Para ello vamos a usar el comando [\\$ZDATETIME](#)

El comando básico sería

```
SET AhoraMismo = $HOROLOG
WRITE AhoraMismo

> 66149,67164
WRITE $ZDATETIME(AhoraMismo)

> 02/09/2022 18:39:24
```

Por defecto, utiliza el formato americano (mm/dd/yyyy). Si quieres utilizar la fecha en otro formato, vamos a utilizar el segundo parámetro, como por ejemplo el europeo (dd/mm/yyyy), en ese caso le vamos a dar el valor 4 (para mas formatos, ver la documentación [\\$ZDATETIME.dformat](#))

```
SET AhoraMismo = $HOROLOG
WRITE AhoraMismo

> 66149,67164

WRITE $ZDATETIME(AhoraMismo,4)

> 09/02/2022 18:39:24
```

Esta opción utiliza como separador y formato de año lo que tengamos definido en las variables locales

Si queremos además poner otro formato la hora, por ejemplo en formato 12 horas (AM/PM) en lugar de formato 24 horas, utilizamos el tercer parámetro con el valor 3, si no queremos que muestre los segundos, usaremos el valor 4 (ver la documentación [\\$ZDATETIME.tformat](#))

```
SET AhoraMismo = $HOROLOG
WRITE AhoraMismo

> 66149,67164

WRITE $ZDATETIME(AhoraMismo,4,3)

> 09/02/2022 06:39:24PM

WRITE $ZDATETIME(AhoraMismo,4,4)

> 09/02/2022 06:39PM
```

¿Ahora está mas claro? pues vamos a profundizar mas

Formato ODBC

Este formato es independiente de tu configuración local, siempre se mostrará como yyyy-mm-dd, su valor es 3. Se recomienda utilizarlo si queremos crear datos que se van a exportar en ficheros, como ficheros CSV, HL7, etc..

```
SET AhoraMismo = $HOROLOG  
WRITE AhoraMismo
```

```
> 66149,67164
```

```
WRITE $ZDATETIME(AhoraMismo,3)
```

```
> 2022-02-09 18:39:24
```

Día de la semana, nombre del día, día del año

Valor	Descripción
10	El día de la semana será un valor entre 0 y 6, siendo el 0 el domingo y el 6 el sábado.
11	El nombre del día de la semana abreviado, lo devolverá según la configuración local definida, la instalación por defecto de IRIS es enuw (English, United States, Unicode)
12	El nombre del día de la semana en formato largo. Igual que 11
14	El día del año, pues eso, el número de días desde el 1 de enero

Si solo queremos tratar fechas y horas por separado, debes de usar los comandos \$ZDATE y \$ZTIME respectivamente. El parámetro para los formatos son los mismos que define en [\\$ZDATETIME.dformat](#) y [\\$ZDATETIME.tformat](#)

```
SET AhoraMismo = $HOROLOG  
WRITE AhoraMismo
```

```
> 66149,67164
```

```
WRITE $ZDATE(AhoraMismo,10)
```

```
> 3
```

```
WRITE $ZDATE(AhoraMismo,11)
```

```
> Wed
```

```
WRITE $ZDATE(AhoraMismo,12)
```

```
> Wednesday
```

¿Y cómo convierto una fecha en formato interno?

Pues ahora vamos a ver el paso contrario, es decir, tener un texto con una fecha y lo convertimos en formato IOS. Para ello vamos a usar el comando [\\$ZDATETIMEH](#)

Esta vez, tenemos que indicar en que formato está la fecha y la hora (si usamos \$ZDATETIMEH) o la fecha (\$ZDATEH) o la hora (\$ZTIMEH) por separado.

Los formatos siguen siendo los mismos, es decir si tenemos una cadena con la fecha en formato ODBC (yyyy-mm-dd), pues utilizaremos el valor 3

```
SET MiFecha = "2022-02-09 18:39:24"
SET FechaInternal = $ZDATETIMEH(MiFecha, 3, 1) // Formato ODBC

SET MiFecha = "09/02/2022 18:39:24"
SET FechaInterna2 = $ZDATETIMEH(MiFecha, 4, 1) // Formato europeo

SET MiFecha = "02/09/2022 06:39:24PM"
SET FechaInterna3 = $ZDATETIMEH(MiFecha, 1, 3) // Formato Americano con hora en 12h A
M/PM
WRITE FechaInternal,!,FechaInterna2,!,FechaInterna3

> 66149,67164
66149,67164
66149,67164
```

Lógicamente, si le decimos que es un formato y le damos el valor erroneo, puede pasar cualquier cosa, como que en lugar de 9 de febrero lo entiende como 2 de septiembre.

No mezclar formatos que luego vienen los problemas.

```
SET MiFecha = "09/02/2022"

/// Formato Americano
SET FechaInterna = $ZDATEH(MiFecha, 1)

/// Formato Europeo
SET OtraFecha = $ZDATETIME(FechaInterna, 4)
WRITE FechaInterna,!,OtraFecha

> 66354
02/09/2022
```

Ni que decir, si intentamos poner fecha europea e intentar transformarlo en americano... ¿Que pasaría en San Valentín?

```
SET MiFecha = "14/02/2022"
SET FechaInterna = $ZDATEH(MiFecha, 1) // Formato Americano. OJO, el mes 14 no existe
^
<ILLEGAL VALUE>
```

Pues como todos los San Valentín.. corazones rotos, bueno... mas bién código roto.

Pues vamos a hacer algo con esto que ya has aprendido

```
READ !,"Por favor, indica tu fecha de nacimiento (dd/mm/yyyy): ",fechaNacimiento
SET formatoInterno = $ZDATEH(fechaNacimiento, 4)
SET diaSemana = $ZDATE(formatoInterno, 10)
SET nombreDia = $ZDATE(formatoInterno, 12)
WRITE !,"El día de la semana de tu nacimiento es: ",nombreDia
IF diaSemana = 5 WRITE "Siempre te ha gustado la fiesta!!!" // Nació un viernes
```

Pero mas adelante veremos otras formas de hacer las cosas, y como controlar los errores.

Próximo capítulo: [Cómo viajar en el tiempo](#)

Curiosidad

Si quieres saber porque se toma el valor de 01/01/1841 como el valor 1, viene porque fué elegido esta fecha por ser el año no bisiesto anterior al nacimiento del ciudadano estadounidense de mayor edad vivo, que era un veterano de la guerra civil con 121 años, cuando se diseñó el [lenguaje de programación MUMPS](#), del cual extiende Object Script

[#Concurso](#) [#Consejos y trucos](#) [#ObjectScript](#) [#Caché](#)

URL de
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-convertirse-en-un-se%C3%B1or-del-tiempo-el-nacimiento>