

Artículo

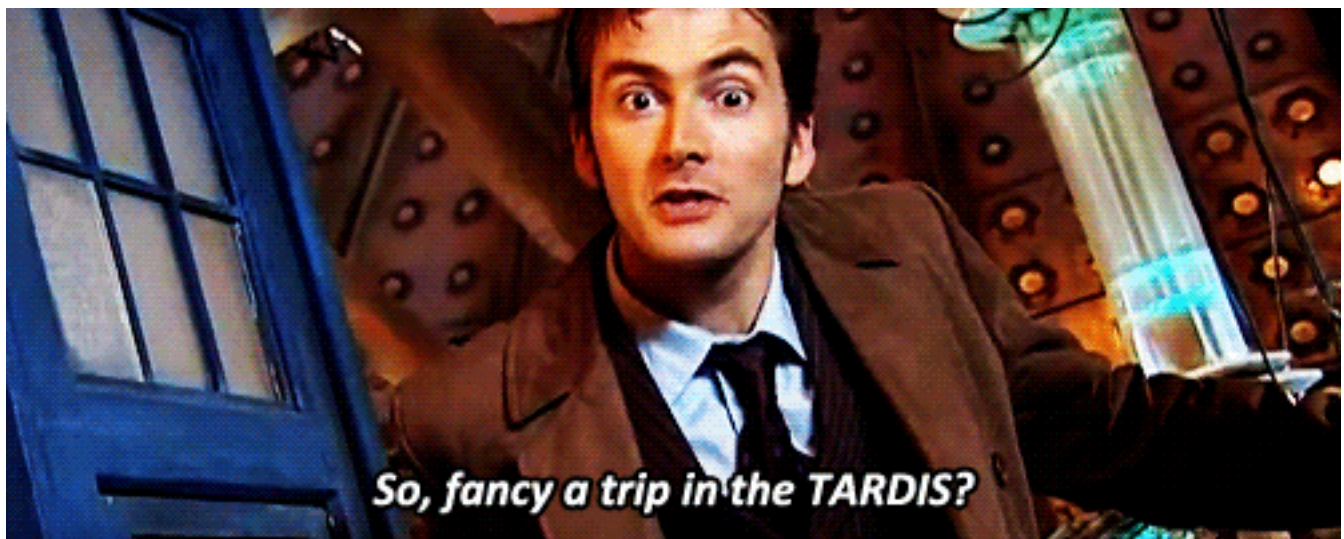
[Kurro Lopez](#) · 14 feb, 2022 Lectura de 4 min

Cómo convertirse en un señor del tiempo - Viajar en el tiempo

Viajar en el tiempo es como visitar París. No puedes simplemente leer la guía, tienes que arrojarte. Come la comida, usa los verbos equivocados, recibe el doble de cargos y terminas besando a completos desconocidos.

El Doctor

Vamos ahora a viajar por el tiempo, osea, vamos a ver fechas futuras y pasadas y como calcularlas en diferentes formatos. La TARDIS no espera, ponte a los mandos y sujétate fuerte.



Vamos al mañana

Como ya vimos en el [anterior artículo](#), las fecha en formato interno se divide en el número de días desde 1 de enero de 1841 y el número de segundos desde las 00:00 horas.

Si queremos aumentar un día, podemos pensar... con que sume un valor o se lo reste a la parte de la fecha... ya estoy viajando por el tiempo.

```
SET Hoy = $ZDATE($HOROLOG,4)
SET FechaInterna = $ZDATEH(Hoy, 4)
SET Futuro = FechaInterna + 2 SET FechaFutura = $DATE(Futuro, 4)
WRITE "Hoy: "_Hoy,!, "FechaInterna: "_FechaInterna,!, "Futuro: "_Futuro,!, "Fecha futura
: "_FechaFutura
```

> Hoy: 02/09/2022

```
FechaInterna: 66149
Futuro: 66151
Fecha futura: 04/09/2022
```

Que fácil.. ya sabes como moverte por los días. Pero si quieres ir al mes siguiente, la cosa cambia. El calendario no tiene el mismo número de días de un mes a otro, así que no vale con sumar 30 días a la fecha. Te presento a nuestra comanda para añadir o restar partes de tiempo.

Si eres usuario de T-SQL verás que el comando te sonará bastante: [DATEADD](#).

Pertenece a la librería \$SYSTEM.SQL.Functions

Regreso al futuro

Podemos añadir meses, días, años.... o quitarlo a la fecha indicada.

```
SET FechaInterna = $ZDATEH("09/02/2022", 4)
SET Futuro = $SYSTEM.SQL.DATEADD("mm",1,FechaInterna)
WRITE "FechaInterna: "_FechaInterna,!, "Futuro: "_Futuro
```

```
FechaInterna: 66149
Futuro: 2022-03-09 00:00:00
```

Si queremos quitar días, meses y años tendríamos que hacerlo por partes, cada uno de los pasos por separado

```
Set FechaInterna = $ZDATEH("09/02/2022", 4)
Set PastDays = $SYSTEM.SQL.DATEADD("dd",12,FechaInterna)
Set PastMonths = $SYSTEM.SQL.DATEADD("mm",8,PastDays)
Set PastYears = $SYSTEM.SQL.DATEADD("yy",-7,PastMonths)
WRITE PastYears
```

```
2015-10-21 00:00:00
```

¿¿Mc Fly, Estás por ahí??

Ya mismo es mi cumpleaños, o no...

Si lo que queremos es saber es cual es la diferencia entre dos fechas, usaremos el comando [DATEDIF](#).

```
Set fechaInterna = $ZDATEH("09/02/2022", 4)
Set miCumple = $ZDATEH("01/02/2023", 4)
Set dias = $SYSTEM.SQL.DATEDIFF("dd",fechaInterna, miCumple)
WRITE !,"Quedan ",dias," días para que sea mi cumpleaños"

> Quedan 357 días para que sea mi cumpleaños
```

Pues mira que bien... ya sabemos como ver la diferencia entre dos fechas por días, meses, años, etc....

Quiero ver a Cristobal Colón descubrir América

Bien, como hemos dicho, nuestro formato de hora tiene el valor 1 para el 01/01/1841, y Cristobal Colón llegó a América el 12/10/1492, entonces pensarás... bueno, voy poniendo valores negativos a la fecha de 1 hasta llegar a 1492

```
WRITE $ZDATE(-1)
```

```
<VALUE OUT OF RANGE>
```

Vaya... Houston, tenemos un problema. No podemos usar valores negativos para obtener una fecha anterior a 1841. Bueno, pues hago la conversión de fecha en texto a formato interno y veo que ha devuelto.

```
WRITE $ZDATEH("12/10/1492",4) // Formato Europeo
```

```
<VALUE OUT OF RANGE>
```

MMMMEEEECCC !!! Gracias por concursar, pero no es la respuesta correcta.
¿Cómo podemos solucionarlo?

Esto es debido a que las fechas tiene definido un valor mínimo que impide ir mas allá, por lo que si utilizamos el parámetro mindata nos permitirá utilizar valores negativos en el formato interno.

```
WRITE $ZDATEH("12/10/1492",4,,,,,-672045)  
> -127184
```

El valor que se ha puesto en la posición 7ª de los parámetros indica que el valor mínimo aceptado será -672045, para saciar tu curiosidad, corresponde a 01/01/0001, además es el mínimo número que podemos utilizar.

Si ves que te pierdes a la hora de contar "comas" para añadir este parámetro, puedes configurar el formato de idioma nacional (National Language Support, NLS) y ya lo dejas activo.

```
SET valorOriginal = ##class(%SYS.NLS.Format).GetFormatItem("DateMinimum")  
WRITE valorOriginal,!  
DO ##class(%SYS.NLS.Format).SetFormatItem("DateMinimum", -672045)  
WRITE $ZDATEH("12/10/1492", 4),!  
DO ##class(%SYS.NLS.Format).SetFormatItem("DateMinimum", valorOriginal) // Volvemos  
a valor mínimo establecido originalmente  
  
> 0  
-127184
```

Entonces, ya puedes viajar por el tiempo sin problemas.

Próximo artículo: El tiempo universal

Agradecimiento

Agradezco a [@Robert Cemper](#) su artículo sobre \$Horolog negativo, que me ha ayudado para completar este artículo y que podeis ver en el siguiente enlace

<https://community.intersystems.com/post/date-dec1840-negative-horolog>

[#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-viajar-en-el-tiempo>