

La mejor forma de representar números en c#

Hoy vamos a ver un post corto, pero creo que es importante, ya que mucha gente desconoce esta característica del lenguaje c#, y no solo eso, sino que veremos una librería que nos hace la vida más fácil para representar números grandes.

Índice

- 1 - Tipos enteros
 - 1.1 - Literales
- 2 - Asignar valores en nuestro código
 - 2.1 - Librería Numerize

1 - Tipos enteros

Todo lo que vamos a ver aquí representa los tipos numéricos enteros, osea positivos y negativos sin decimales, en términos del lenguaje C# Tenemos `sbyte` , `byte` , `short` , `ushort` , `int` , `uint` , `long` , `ulong` , `nint` , `nuint` .

La diferencia entre `long` y `ulong` es que `long` tiene signo (positivo o negativo) y `ulong` no, lo que quiere decir que el número que puedes representar es “mas grande”, aunque siempre sera “positivo”) ya que en la versión con signo `32 bits` son positivos y 32 negativos, mientras que en la versión sin signo son `64 bits` positivos.

1.1 - Literales

Dentro de los tipos enteros también nos entran diferentes tipos, que son, los decimales, que los acabamos de ver, los valores hexadecimales, que tienen un prefijo de `0x` , o los valores binarios, que tienen un prefijo de `0b` ;

Por ejemplo si representamos el valor 2022 sería lo siguiente:

```
var hex = 0X7E6
var binary = 0b11111100110
```

2 - Asignar valores en nuestro código


El punto principal por el que creo este post es por la siguiente información. Cuando representamos un número como 2022 es fácil de leer en el código, pero qué pasa si nuestro número es más grande? por ejemplo `43209412016` ; ese número a simple vista es imposible de saber cual es.

C# nos provee de una característica que nos permite representar los números de una forma más sencilla, la cual es añadir la barra baja (`_`) entre números. entonces el siguiente número quedaría de la siguiente manera.

```
long valor = 43_209_412_016
```

2.1 - Librería Numerize

Pero para mí, esto no es suficiente, he creado una librería que nos permite indicar ese mismo número pero en inglés, lo cual, lo hace mucho más fácil de leer.

La librería en cuestión es **Numerize** y te invito a que le des una buena estrella  . Por supuesto está en **Nuget**.

Con la librería podemos escribir el número en inglés sin preocuparnos de barras bajas o de no saber a simple vista qué número es.

Por ejemplo con el número anterior, podemos hacer lo siguiente:

```
long value = Numerize.Fourty.Three().Billion()  
                .Two().Hundred().Nine().Milion()  
                .Four().Hundred().Twelve().Thousand()  
                .Sixteen();
```

Y con el uso de **operadores implícitos** también convierte el número a texto:

```
Numerize numerize = new Numerize();  
string result = numerize.Fourty().Three().Billion()  
                .Two().Hundred().Nine().Milion()  
                .Four().Hundred().Twelve().Thousand()  
                .Sixteen();
```

Y nos lo convierte en lo siguiente (en inglés): `forty three billion two hundred nine million four hundred twelve thousand sixteen` .

Lo cual es mucho más amigable