

Ejercicios:

1. Indica cuáles de los siguientes identificadores son válidos en Visual C#, y explica por qué en aquellos que no lo sean (los identificadores vienen separados por comas):

num, num_valido, num contador, declaracion renta,
declaracion_renta, hoy, mañana, ayer, 1a_escalera, ley22/14, num_2,
cadena, carácter, while, Nombre, NOMBRE

2. Seleccionar el tipo de datos más adecuado para los siguientes conceptos.

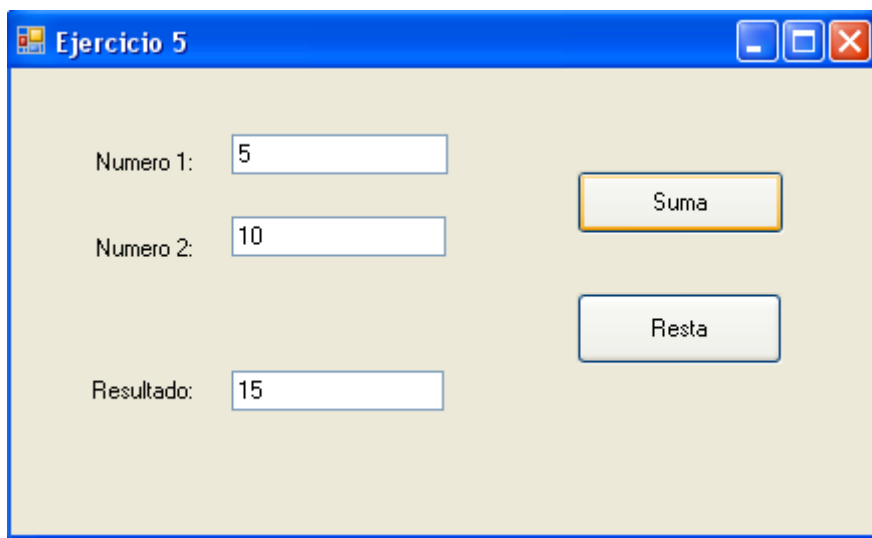
- Dirección.
- Cantidad de un préstamo.
- Número de teléfono.
- Tipo de interés.
- Cumpleaños.
- DNI.
- Nombre de una persona.

3. Analiza qué valores van tomando las variables tras cada una de estas expresiones. Antes de ello indica de qué tipo deberían ser cada una de las variables y decláralas:

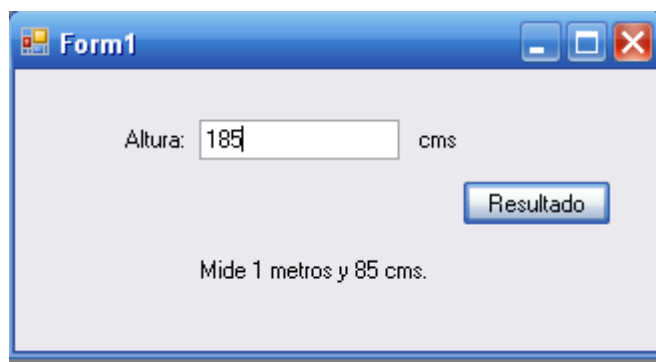
```
num1 = 10;  
num2 = 20;  
num3 = num1 + num2;  
num3 = num3 + 1;  
num3 ++;  
  
r1 = num1 / num2;  
r1 = (double)num1 / (double)num2;  
r2 = 2.0;  
r3 = r2 / r1 - 1;  
r3 = r2 / (r1 - 1);  
  
l1 = num1 > num2;  
l1 = (num1 > num2) && (num2 <= 20);  
l1 = (num1 > num2) || (num2 <= 20) ;  
l1 = (num1 < num2) || (num2 <= 20) && (num1 == 10);  
l1 = ((num1 < num2) || (num2 <= 20)) && (num1 != 10);  
l1 = !((num1 < num2) || (num2 <= 20)) && (num1 != 10);
```

4. Realizar un programa que, en el evento click de un botón, declare dos variables de tipo entero, las inicialice y las sume introduciendo el resultado en otra variable. Imprimir con un MessageBox el resultado de la suma.

5. Realizar un programa que **lea** dos valores de tipo entero y muestre por pantalla el valor de la suma y de la resta de esas dos variables.
Vamos a utilizar try catch en la suma y no en la resta para comparar qué ocurre si el usuario no introduce datos correctos.
Utilizar un formulario similar al siguiente y try catch para evitar el error de introducción de enteros (**Ejercicio Resuelto**):

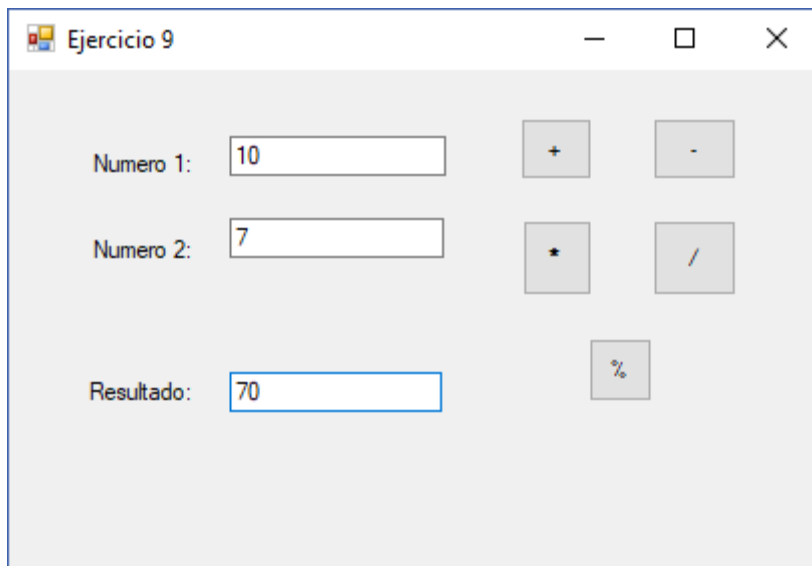


6. Realizar un programa que lea el valor de dos variables de tipo entero y muestre por pantalla con un MessageBox el valor de la división entera y del resto de la división entera.
7. Realizar un programa que lea la altura en cms de un individuo y muestre, en un label, cuántos metros y cms tiene. (185 -> 1 metro, 85 cms. 205-> 2 metros 5 cms.).
Utilizar try – catch para la introducción del número. Realizar con un formulario de la siguiente forma:

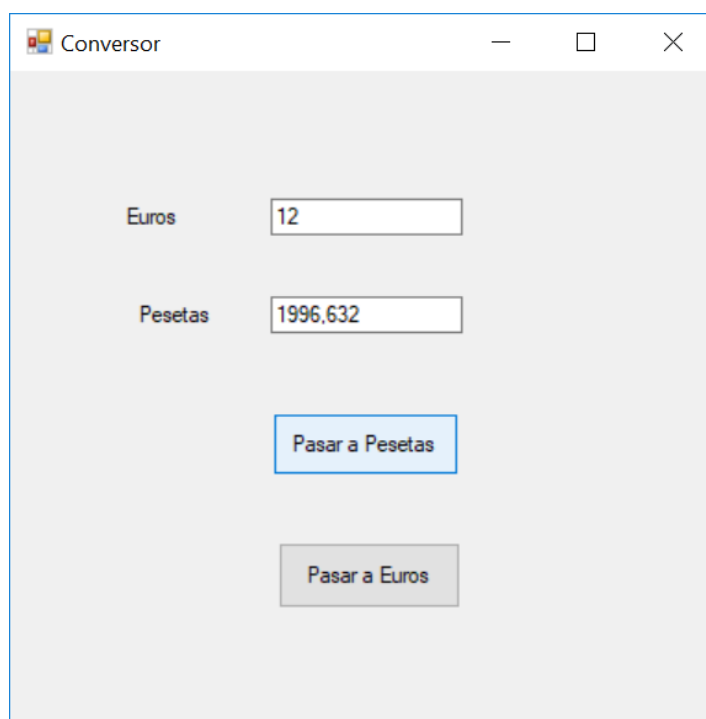


8. Realizar un programa que lea 3 números y calcule su suma y producto almacenando el resultado en dos variables. Mostrar el resultado obtenido por pantalla.

9. Ampliar el **ejercicio 5** de manera que tengamos botones para los 5 operadores aritméticos estudiados en el tema. Utilizar try-catch:



10. Realizar un programa que cambie de pesetas a euros y viceversa. Utilizar try catch en la introducción de los valores:



Utilizar try catch para evitar errores en la introducción de datos por parte del usuario.

11. Dada una cantidad ingresada en un banco por un cliente, y el interés anual, calcular el capital que tendría a final del año.
12. En una tienda se adquieren tres productos. Introduciendo el precio de cada uno de los productos, mostrar por pantalla el importe total a abonar, sin IVA y con IVA, siendo éste del 21%.
13. Un trabajador a lo largo del mes trabaja un número de horas normales y un número de horas extras. Las horas extras se pagan el doble que las horas normales. Realizar un programa en el que se introduzcan el número de horas trabajadas (normales y extras) y la paga por hora normal, y se calcule la nomina mensual, aplicando una retención del 18%.
14. Realizar un programa para probar los distintos operadores relacionales. Al pulsar el botón correspondiente nos aparecerá en un label True o False, como consecuencia de comparar con ese operador relacional el número 1 con el número 2:

