

Enunciado:

Tema: **Excepciones.**

Desarrolle el método indicado en el prototipo de la **Figura 1**, funcionalmente será responsable de realizar la división en dos partes iguales y enteras del número entero pasado como parámetro, debido a esto el parámetro deberá ser un número par. Si el valor del parámetro no es un número par, el método lanzará una excepción del tipo **System.ArgumentException**, indicando que se ha violado la precondition necesaria para realizar la división por dos.

Como podemos notar por su namespace, la excepción **System.ArgumentException** es provista por el .NET Framework o en otras palabras no es una excepción personalizada (custom exception).

Figura 1

```
public static int DividirPorDos(int numeroPar)
```

Una vez desarrollado el método, deberemos desarrollar algunas pruebas para verificar el correcto funcionamiento del mismo. El código de la **Figura 2**, muestra una de las tantas pruebas posibles, quedando el desarrollador en libertad de realizar más y deviseras.

EL funcionamiento de la prueba cuyo código podemos ver en la **Figura 2**, es el siguiente: En la línea (40) se invoca al método a probar pasándole un valor impar, un 15, el resultado esperado para el éxito de la prueba es que el método lance una excepción del tipo **ArgumentException**, razón por la cual pintamos el label llamado "labelPruebaB" de verde (45), el resto de los casos deben indicar una falla del método y por eso pintamos el label llamado "labelPruebaB" de rojo (41, 49), de esta manera podemos ver el resultado de cada prueba en la interfaz de usuario de la **Figura 3**, las pruebas serán ejecutadas cuando se realice un click sobre el botón, siendo el código de atención al evento mostrado en la **Figura 4**. De esta manera hemos automatizado las pruebas que verifican el correcto funcionamiento de nuestro método. Como un experimento, podemos alterar el comportamiento del método y notar como algunas de las pruebas se establecen en color rojo, por ejemplo comentando la línea que dispara la excepción **ArgumentException**.

Figura 2

```

36 private void EjecutarPruebaB()
37 {
38     try
39     {
40         int resultado = Divisor.DividirPorDos(15);
41         labelPruebaB.ForeColor = Color.Red;
42     }
43     catch (ArgumentException)
44     {
45         labelPruebaB.ForeColor = Color.Green;
46     }
47     catch (Exception)
48     {
49         labelPruebaB.ForeColor = Color.Red;
50     }
51 }

```

Figura 3

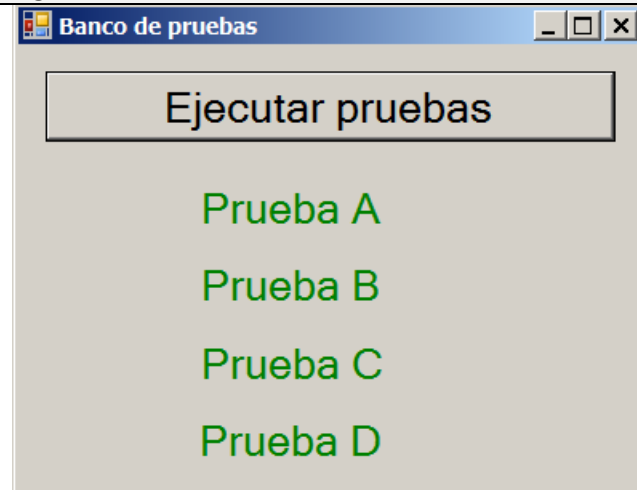


Figura 4

```

19 private void buttonEjecutarPruebas_Click(object sender, EventArgs e)
20 {
21     EjecutarPruebaA();
22     EjecutarPruebaB();
23     EjecutarPruebaC();
24     EjecutarPruebaD();
25 }

```