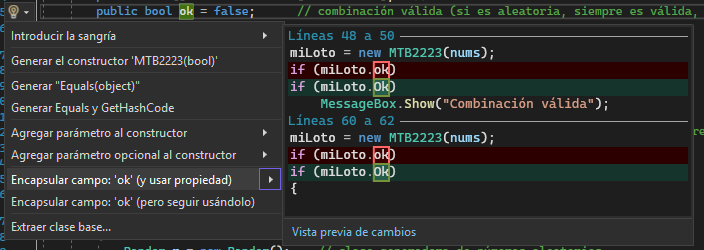
* Encontrar cinco errores de normas de estilo en el fichero *loto.cs,*indicando número de línea, error encontrado y solución.
  1. Llaves (líneas 77 a 79): las llaves deben estar incluidas en el código (aunque el compilador no lo detecta como error). Se deben incluir incluso si hay una sola línea. La solución sería añadir llaves.
  2. Salto de línea (línea 79): el if y el bloque interior deberían estar en líneas distintas.
  3. Nombres no autodescriptivos (líneas 76 y 74): premi y a deberían tener nombres mas sencillos que no necesiten un comentario para especificar qué es, les cambio el nombre a combinacionGanadora y aciertos respectivamente.
  4. Espaciado (líneas 77 y 78): después de punto y coma y entre operadores debería haber espacios. Como excepción se pueden dejar juntos los operadores unarios como i++. Se escribiría correctamente: for (int i = 0; i < MAX\_NUMEROS; i++)
  5. Nomenclatura (línea 74): el método comprobar debería seguir la nomenclatura PasCal, no camelCase, por lo que le cambio el nombre como en las anteriores; usando Ctrl + R, R a Comprobar. De esta forma se actualiza el nombre seleccionado y todas sus referencias.

Estos errores no son los únicos que he encontrado, se repiten estos mismos y aparecen otros como el de tabulación de la línea 75 pero señalo estos únicamente. (En las líneas 14 y 15 los nombres podrían ser más descriptivos también)

* Si existen, detectar y aplicar al menos tres patrones de refactorización (tanto en el fichero *Loto.cs* como en el fichero *Form1.cs*), indicando el patrón que se y, si es posible aplicarlo con Visual Studio, la opción que se usa.
  1. Numero mágico: el numero 6 aparece muchas veces en el Form1.cs haciendo referencia al máximo de números (igual que en el loto.cs). Visual studio no ofrece posibilidad de refactorización automática por el momento en este patrón.
  2. Método largo (en este caso el constructor del formulario): dedica 6 lineas con dos puntos y coma cada una y hace que el constructor no se quede limpio, por lo que extraemos el código del constructor. Visual studio nos permite hacerlo automáticamente con una opción llamada extraer método.
  3. Encapsulacion: la línea 15 de loto.cs contiene un miembro publico y siguiendo el principio de encapsulación de la POO deberían ser privados. En caso de que se necesite acceder a ellos desde fuera se crea una propiedad. Visual studio nos permite una refactorización automática 

Nótese que no solo genera la propiedad y la usa, si no que el propio miembro cambia a private:

