Examen 2EV JPS 22/23

JORGE PARDO SEMPERE - DAW

1)

Linea 28: Codigo desordenado, para mejorar su lectura podríamos declarar las variables en una línea y luego inicializar la que necesitemos:

```
int i, j, num;
i = 0;
```

```
56      public Loteria(int[] misnums)  //
```

Linea 56: Renombramos misnums con caMel

```
75      public int comprobar(int[] premi)
76      {
77          int aciertos=0;                    // número de aciertos
78          for (int i=0; i<MAX_NUMEROS; i++)
79              for (int j=0; j<MAX_NUMEROS; j++)
80                  if (premi[i]==Nums[j]) aciertos++;
```

Linea 75: El nombre del método debe ir en Pascal, cambiamos por “Comprobar”

Linea 77: Añadimos espaciados para mejorar la legibilidad del código.

Código después de aplicar normas de estilo, mucho más legible.

```
1 referencia | 0 cambios | 0 autores, 0 cambios
public int Comprobar(int[] premi)
{
    // número de aciertos
    int aciertos = 0;

    for (int i=0; i<MAX_NUMEROS; i++)
    {

        for (int j = 0; j < MAX_NUMEROS; j++)

            if (premi[i] == Nums[j])
            {
                aciertos++;
            }
    }
```

3)

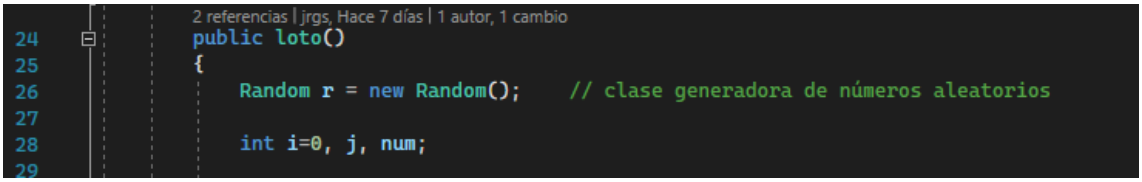
```
81          return a;
```

Linea 81: RENOMBRAR. Nombre poco descriptivo, cambiamos a “aciertos” para saber que es el valor que nos devuelve.

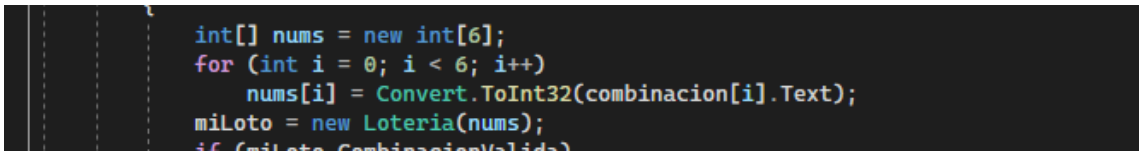
Seleccionamos la variable a Renombrar, Ctrl R + Ctrl R, y Visual Studio nos permite renombrarla cambiándola en todas las instancias que aparece si así lo queremos.

```
15          public bool ok = false;       // combinación válida
```

Linea 15: RENOMBRAR. Nombre poco descriptivo, cambio por CombinacionValida.



Línea 26: Random r, nombre poco descriptivo, podríamos cambiarlo por “random”.



Código que se reptite en varios puntos de la clase, hacemos una Extracción de método, Visual Studio nos permite hacerlo (Editar>Refactorizaciones>Extraer método

Podríamos usar try-catch y lanzar excepciones para el manejo de datos no validos de entrada, así como el caso de que nos introduzcan menos números, lo que provoca una excepción no controlada.

4) Diseño de prueba

El constructor tiene dos posibles valores de salida, true o false. Hay dos posibilidades de error: que el número introducido sea menor que 1, o mayor de 49.

(También hay una excepción no controlada si no se introducen todos los valores)

Números >= 1 y <= 49 = True

Números menores 1 = False

Números mayores 49 = False

Valores frontera a probar: 0,1,49,50