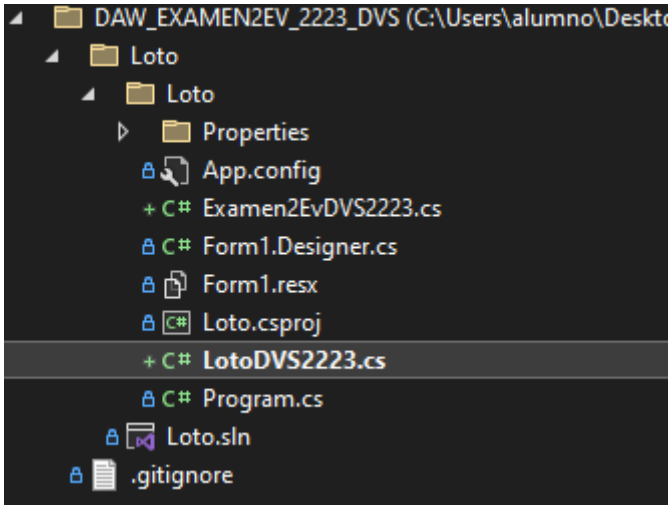
EXAMEN 2ª EVALUACIÓN:

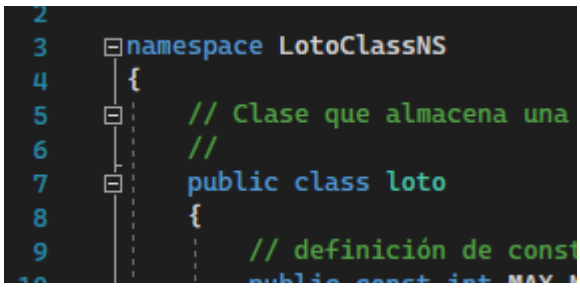
ENTORNOS DE DESARROLLO

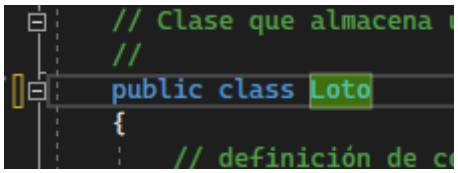
En el primer punto, comenzamos cambiando el nombre a la clase Form.cs y a Loto.cs, cómo método de autenticación…



EJERCICIO 1: Buscando errores.

En primer lugar, en la línea 7, cuando declaramos la clase loto, esta debería estar en PasCal, siendo Loto. Hacemos click derecho -> Cambiar nombre, o el shortcut doble Ctrl+R.





Línea 14, el vector nums, estaría mejor llamándose numeros, en caMel, de forma más clara y sin barra baja. Cambiamos nombre.

```
14        private int[] _nums = new int[MAX_NUMEROS];
15        public bool ok = false;       // combinación vá
16
```

```
14        private int[] numeros =
15        public bool ok = false;
```

Linea 17, el get y el set deberían estar en corchetes distintos…

```
16
17        public int[] Numeros {  //DVS2223
18            get => numeros;
19            set => numeros = value;
20        }
21
22        // En el caso de que el constructo
```

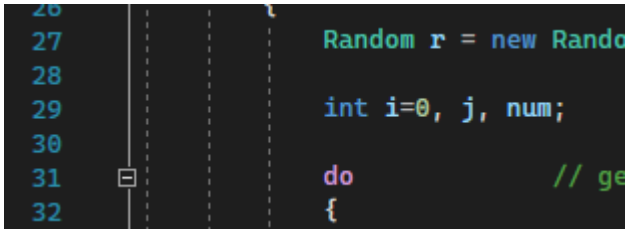
```
16
17        public int[] Numeros {  //DVS2223
18
19            { get => numeros; }
20            { set => numeros = value; }
21        }
22
```

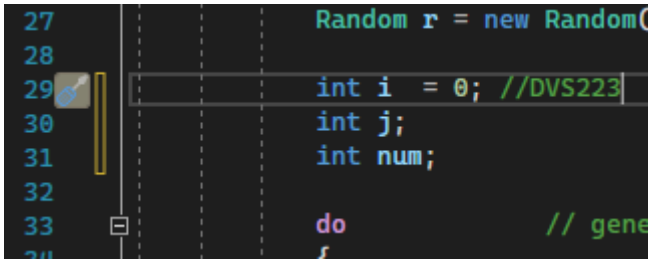
En la línea 25, el constructor de la clase debe tener el nombre en PasCal, o en todo caso idéntico al de la clase.

```
22
23        // En el caso de que
24        //
25        public Loto()
26        {
27            Random r = new R
28
```

```
23        // En el caso de que el
24        //
25        public Loto()
26        {
27            Random r = new Randor
28
29            int i=0, j, num;
30
31            do              // ge
```

En la línea 29, deberíamos declarar las variables a poder ser una por línea, y nada de inicializar una variable seguida de una declaración. Además, poner espacios entre el asignador “=”.



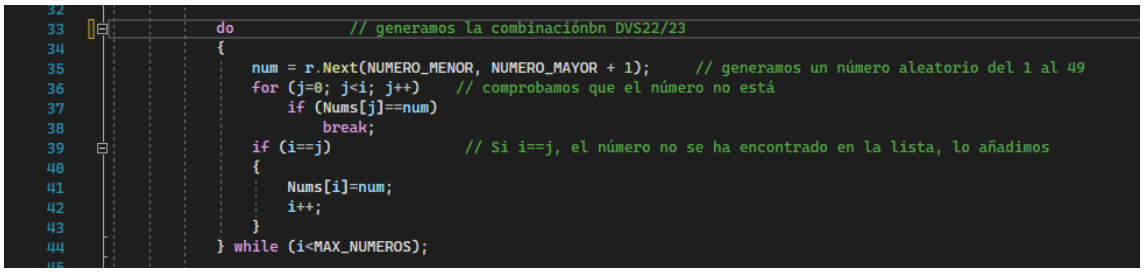


Linea 35, dejar espacio entre declaración y bucle.

Linea 36, no hay espacios entre comparadores y asignacion.

Linea 37, abrir llaves para sentencias del bucle.

Linea 39, no hay espaciado entre el comparador y las variables.



Linea 54, clase loto debe estar en PasCal.

Linea 56, lo de siempre de los espacios…

Linea 56, parece que estaba rota la barra espaciadora.

Línea 58, tratamos de separar declaración de bucle.

Linea 59, espacios.

Linea 60, espacios y debe ir entre llaves por pertenecer al for

Linea 61, faltan llaves para las sentencias del if.

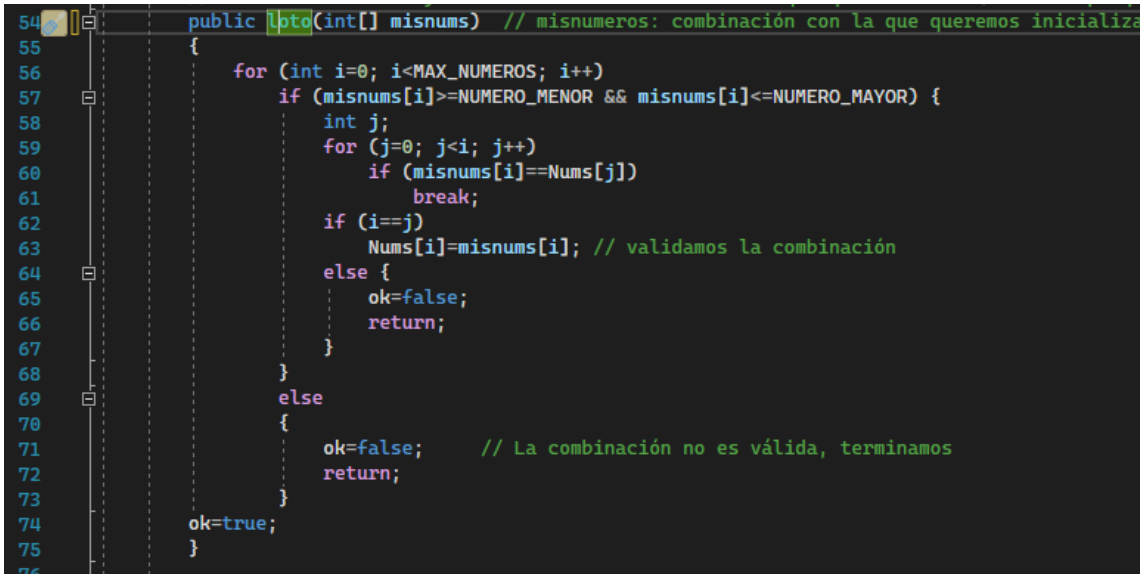
Línea 62, espacios.

Linea 63, entre llaves las sentencias del if.

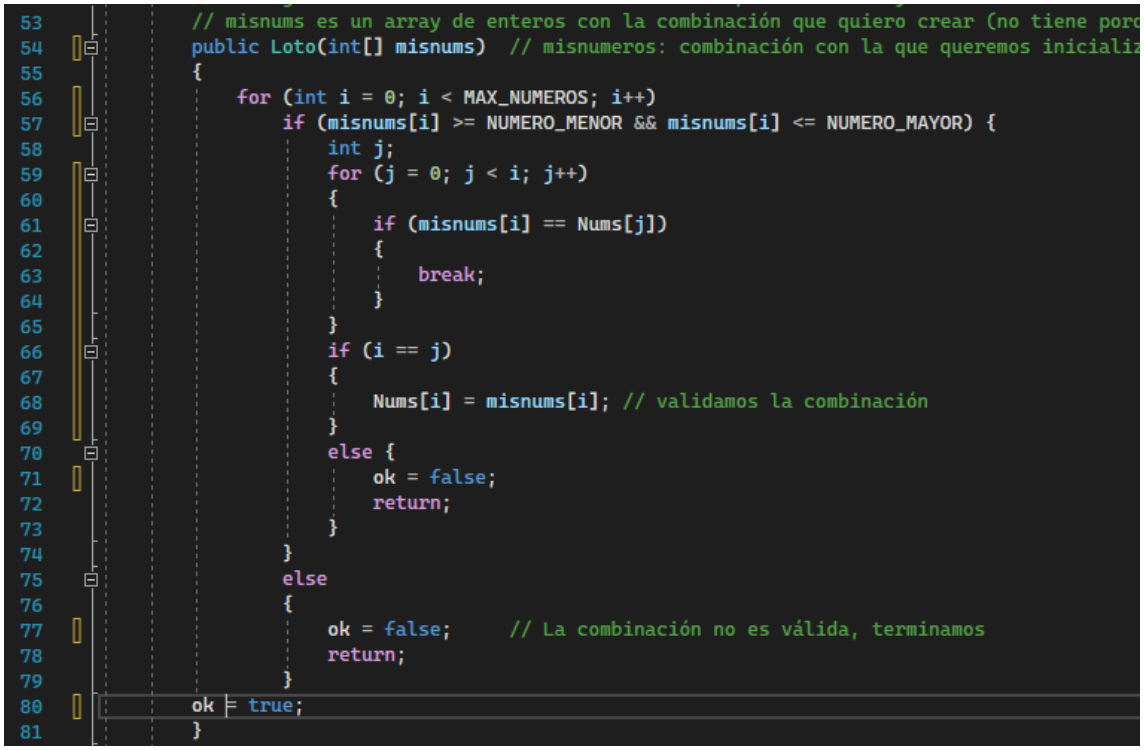
Linea 65, espacios.

Linea 71, espacios.

Linea 74, espacios.



```
54      public loto(int[] misnums)  // misnumeros: combinación con la que queremos inicializa
55      {
56          for (int i=0; i<MAX_NUMEROS; i++)
57              if (misnums[i]>=NUMERO_MENOR && misnums[i]<=NUMERO_MAYOR) {
58                  int j;
59                  for (j=0; j<i; j++)
60                      if (misnums[i]==Nums[j])
61                          break;
62                  if (i==j)
63                      Nums[i]=misnums[i]; // validamos la combinación
64                  else {
65                      ok=false;
66                      return;
67                  }
68              }
69              else
70              {
71                  ok=false;     // La combinación no es válida, terminamos
72                  return;
73              }
74          ok=true;
75          }
```

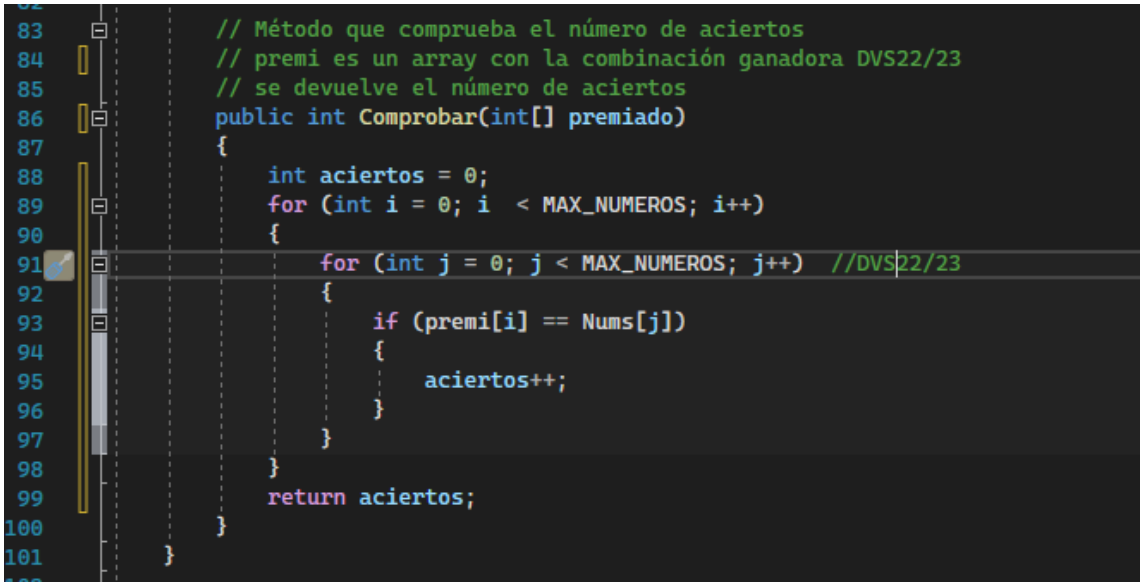


```
53      // misnums es un array de enteros con la combinación que quiero crear (no tiene porq
54      public Loto(int[] misnums)  // misnumeros: combinación con la que queremos inicializ
55      {
56          for (int i = 0; i < MAX_NUMEROS; i++)
57              if (misnums[i] >= NUMERO_MENOR && misnums[i] <= NUMERO_MAYOR) {
58                  int j;
59                  for (j = 0; j < i; j++)
60                  {
61                      if (misnums[i] == Nums[j])
62                      {
63                          break;
64                      }
65                  }
66                  if (i == j)
67                  {
68                      Nums[i] = misnums[i]; // validamos la combinación
69                  }
70                  else {
71                      ok = false;
72                      return;
73                  }
74              }
75              else
76              {
77                  ok = false;     // La combinación no es válida, terminamos
78                  return;
79              }
80          ok = true;
81          }
```

Linea 86, el método debe ir en PasCal, y el parámetro describirse de un modo más claro, como por ejemplo *premiado*.

Línea 88, además de la ya recurrente falta de espacios, la variable aciertos debe definirse de forma más clara, para evitar confusiones y descripciones innecesarias.

De la línea 89 a la 92, falta todos los espacios adyacentes a las asignaciones y comparaciones, además de falta de llaves. Además, en la línea 91 en cocncreto falta un salto de línea además de la ya mencionada falta de llaves. Ultima corrección…



```
83          // Método que comprueba el número de aciertos
84          // premi es un array con la combinación ganadora DVS22/23
85          // se devuelve el número de aciertos
86          public int Comprobar(int[] premiado)
87          {
88              int aciertos = 0;
89              for (int i = 0; i  < MAX_NUMEROS; i++)
90              {
91                  for (int j = 0; j < MAX_NUMEROS; j++)  //DVS22/23
92                  {
93                      if (premi[i] == Nums[j])
94                      {
95                          aciertos++;
96                      }
97                  }
98              }
99              return aciertos;
100         }
101     }
```

EJERCCIO 3: Diseño de pruebas de caja negra.

Para ello, vamos a fijarnos en los posibles inputs, para intentar establecer el máximo cubrimiento.

Debemos realizar las pruebas al constructor con el parámetro, el cual es el de nuestros números elegidos. Vamos a intentar cubrir todos los casos eligiendo los valores frontera adecuados.

CASO	CLASE DE EQUIVALENCIA	VALORES ELEGIDOS	RESULTADO ESPERADO
A1	Los 6 números son correctos.	1 2 3 4 5 6	Validamos la combinación.
A2	Todos los números correctos menos 1	1 2 3 4 5 5	La combinación no es válida.
A3	Ningún número correcto	-1 -1 -1 -1 -1 -1	La combinación no es válida.
A4	Solo un número es correcto	1 50 50 50 50 50	La combinación no es válida.

No encuentro ningún error en el planteamiento anterior tras realizar las pruebas de caja negra. Tampoco dispongo ahora mismo de la forma clara para realizar pruebas unitarias a un constructor. En la práctica que realicé, aprendí a realizar pruebas unitarias a métodos de clases, pero no sé cómo plantearlo para un constructor… Te he creado el proyecto de pruebas unitarias…Espero tener contenido junto a las memorias aunque sea para un cinco, prometo mejorar para la tercera evaluación… Un saludo.