

UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

Reporte Laboratorio 7

Ingeniería de Software

Profesor: Allan Berrocal

Jason Murillo Madrigal, B54956

I semestre, 2023

Archivo:	BosqueEncantado.cs	
Método	# Línea(s)	Tipo de error y descripción
Bosque Encantado	12-15	Se utilizan nombres de variables no significativos como pfilas o pcolumnas.
Bosque Encantado	23	En C# no existe Math.random
Bosque Encantado	25	Se puede sustituir int_decision y colocar directamente los valores y no crear variables extra.
Bosque Encantado	23	doubleAleatorio no tiene una función clara, cuando se declara y se asigna en la línea 23 y luego se vuelve asignar en la línea 40, se asignan de manera innecesaria.
num_lag_vec	47	Método con demasiados parámetros.
num_arb_vec	66	Método con demasiados parámetros.
Todos	1-124	El código no tiene comentarios.
Bosque Encantado	24	numeroAleatorio no fue declarado.
Bosque Encantado	25	int_desicion no esta declarado en ese método.

Archivo:	Hangman.cs	
Método	# Línea(s)	Tipo de error y descripción
Class HangMan	6	Debería seguir la convención e iniciar cada palabra en mayúscula
Perdio	61-78	Método comentado sin ninguna función o utilidad.
readdata	89	No se utiliza en otras partes del código.
readdata	91	FileStream debería utilizar la instrucción using.
PlayHangman	15	Variable con nombre poco significativo "idx".
PlayHangman	46	Variable con nombre poco significativo "a".
PlayHangman	35	No se están manejando excepciones.
Todos	1-106	El código no tiene comentarios.

Archivo:	Letters.cs	
Método	# Línea(s)	Tipo de error y descripción
Count	17-32	Se podría optimizar la forma en que se analiza si tiene una letra para que sea más legible y simple.
Count	24-31	No se está calculando correctamente las consonantes, se están tomando en cuenta otras letras.

Todo el código	1-40	No está bien indentado, espacios diferentes de sangría.
Count	10	No se están manejando excepciones.
Count	9-11	No se están haciendo validación de entrada de datos.
Todos	1-41	El código no tiene comentarios.
Count	35-37	No se está haciendo validación de entrada de datos.

Archivo:	NumberGuessing.cs	
Método	# Línea(s)	Tipo de error y descripción
play	11	Debería seguir la convención e iniciar cada palabra en mayúscula y tener un nombre significativo.
Todo el código	1-53	No está bien indentado, espacios diferentes de sangría.
play	15	No se están manejando excepciones.
NumberGuessing	44-46	Comentarios que no son necesarios.
Todos	1-106	El código no tiene comentarios.

Archivo:	Pantalla.cs	
Método	# Línea(s)	Tipo de error y descripción
public class Pantalla	1	El nombre “pantalla” no es bastante descriptivo.
public class Pantalla	3	El nombre “matriz” no es bastante descriptivo
public class Pantalla	3	El tipo de dato var no es correcto porque no permite saber con exactitud el tipo de dato de la matriz.
public class Pantalla	6	El nombre “crear” no es bastante descriptivo.
Crear	8	GetLenght esta mal escrito, lo correcto sería GetLength
Crear	9	GetLenght esta mal escrito, lo correcto sería GetLength
Crear	10	No se está concatenando, se está sobrescribiendo en cada iteración.

Archivo:	Program.cs	
Método	# Línea(s)	Tipo de error y descripción
Class Program	3	El nombre "program" no es bastante descriptivo.
Class Program	5-10	Comentarios no significativos o descriptivos.
Main	14-17	Se comentaron los llamados a los métodos por lo cual no se ejecuta el código.
Todos	1-20	El código no tiene comentarios.

Archivo:	Roman.cs	
Método	# Línea(s)	Tipo de error y descripción
Todo	1-101	No está bien indentado, espacios diferentes de sangría.
public class Pantalla	7	El nombre "Roman" no es bastante descriptivo.
Todos	1-101	El código no tiene comentarios.
Combine	43	Val1 y Val2, nombres poco significativos.
Combine	43	Se espera que Val1 y Val2 sean tipo entero

Combine	48	Se tiene que especificar más la excepción.
To	55	El nombre "To" no es bastante descriptivo.

Solución de errores			
Archivo	Método	# Línea(s)	Tipo de error y solución
BosqueEncantado	Bosque Encantado	23	<p>En C# no existe Math.random</p> <p>Se propone la siguiente solución: Utilizar Random()</p>

Antes

```
int int_decision;
for (int f = 0; f < cantidadFilas; ++f)
{
    for (int c = 0; c < cantidadColumnas; ++c)
    {
        doubleAleatorio = (Math.random() * 3) + 1;
        int_decision = (int)numeroAleatorio;
        if (int_decision == 1)
        {
            matrizBosque[f,c] = 1;
        }
        if (int_decision == 2)
        {
            matrizBosque[f,c] = 2;
        }
    }
}
```

Después

```
0 references
public BosqueEncantado(int pfilas, int pcolumnas)
{
    cantidadFilas = pfilas;
    cantidadColumnas = pcolumnas;
    matrizBosque = new int[cantidadFilas, cantidadColumnas];
    double doubleAleatorio;
    int int_decision;
    for (int f = 0; f < cantidadFilas; ++f)
    {
        for (int c = 0; c < cantidadColumnas; ++c)
        {
            doubleAleatorio = (new Random().NextDouble() * 3) + 1;
            int_decision = (int)numeroAleatorio;
            if (int_decision == 1)
            {
                matrizBosque[f,c] = 1;
            }
            if (int_decision == 2)
            {
                matrizBosque[f,c] = 2;
            }
        }
    }
}
```

Se deberían utilizar cosas que si existen en el lenguaje, se propone cambiar a una función que si existe en este lenguaje. Una función que no es del lenguaje puede ocasionar errores, se usara new random en vez de Math. random

BosqueEncantado	Bosque Encantado	24	Se debería utilizar doubleAleatorio. en vez de numeroAleatorio.
-----------------	------------------	----	-----------------------------------------------------------------

Antes

```

0 references
public BosqueEncantado(int pfilas, int pcolumnas)
{
    cantidadFilas = pfilas;
    cantidadColumnas = pcolumnas;
    matrizBosque = new int[cantidadFilas, cantidadColumnas];
    double doubleAleatorio;
    int int_decision;
    for (int f = 0; f < cantidadFilas; ++f)
    {
        for (int c = 0; c < cantidadColumnas; ++c)
        {
            doubleAleatorio = (Math.random() * 3) + 1;
            int_decision = (int)numeroAleatorio;
            if (int_decision == 1)
            {
                matrizBosque[f,c] = 1;
            }
            if (int_decision == 2)
            {

```

Después

```

0 references
public BosqueEncantado(int pfilas, int pcolumnas)
{
    cantidadFilas = pfilas;
    cantidadColumnas = pcolumnas;
    matrizBosque = new int[cantidadFilas, cantidadColumnas];
    double numeroAleatorio;
    int int_decision;
    for (int f = 0; f < cantidadFilas; ++f)
    {
        for (int c = 0; c < cantidadColumnas; ++c)
        {
            numeroAleatorio = (new Random().NextDouble() * 3) + 1;
            int_decision = (int)numeroAleatorio;
            if (int_decision == 1)
            {
                matrizBosque[f,c] = 1;
            }
            if (int_decision == 2)
            {

```

Se corrige el doubleAleatorio por numeroAleatorio, que también se declara arriba. El problema radica en que no se declaró numeroAleatorio y se estaba utilizando, además que se estaba declarando otra variable que no se estaba utilizando(doubleAleatorio).

HangMan	PlayHangman	15	Variable con nombre poco significativo "idx".
---------	-------------	----	-----------------------------------------------

Antes

```

0 references
public static void PlayHangman()
{
    Console.WriteLine("Welcome to Hangman!!!!!!!!!!");

    StreamReader file = new StreamReader("usa.txt");
    Random randGen = new Random();
    var idx = randGen.Next(0, 61333);
    //var idx = randGen.Next(0, 10);
    string mysteryWord = new string("");
    for (int i = 0; i < idx; i++) {
        mysteryWord = file.ReadLine();
    }
    file.Close();

    char[] guess = new char[mysteryWord.Length];
    Console.Write("Please enter your guess: ");
}

```

Después

```

0 references
public static void PlayHangman()
{
    Console.WriteLine("Welcome to Hangman!!!!!!!!!!");

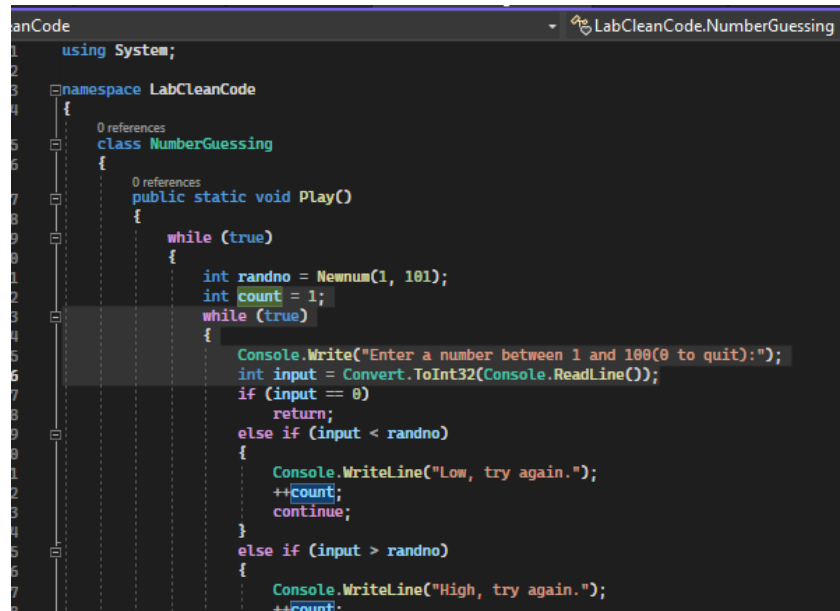
    StreamReader file = new StreamReader("usa.txt");
    Random randGen = new Random();
    var randomNumber = randGen.Next(0, 61333);
    //var idx = randGen.Next(0, 10);
    string mysteryWord = new string("");
    for (int i = 0; i < idx; i++) {
        mysteryWord = file.ReadLine();
    }
    file.Close();
}

```

Utilizar variables con nombres no significativos, es una mala practica de programación. Se propone utilizar un nombre mas significativo como randomNumber.

Todos	1-106	El código no tiene comentarios que describan la función de cada método.
-------	-------	-------------------------------------------------------------------------

Antes

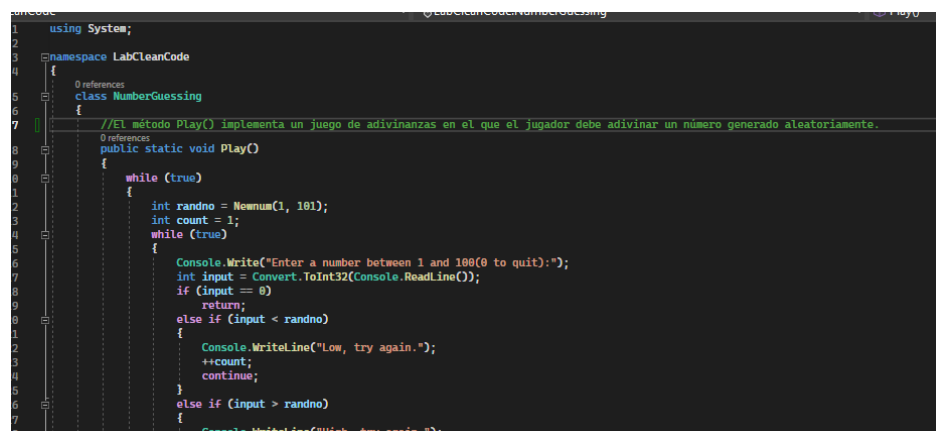


```

1 using System;
2
3 namespace LabCleanCode
4 {
5     0 references
6     class NumberGuessing
7     {
8         0 references
9         public static void Play()
10        {
11            while (true)
12            {
13                int randno = Newnum(1, 101);
14                int count = 1;
15                while (true)
16                {
17                    Console.Write("Enter a number between 1 and 100(0 to quit):");
18                    int input = Convert.ToInt32(Console.ReadLine());
19                    if (input == 0)
20                        return;
21                    else if (input < randno)
22                    {
23                        Console.WriteLine("Low, try again.");
24                        ++count;
25                        continue;
26                    }
27                    else if (input > randno)
28                    {
29                        Console.WriteLine("High, try again.");
30                        ++count;
31                    }
32                }
33            }
34        }
35    }
36 }

```

Después



```

1 using System;
2
3 namespace LabCleanCode
4 {
5     0 references
6     class NumberGuessing
7     {
8         0 references
9         //El método Play() implementa un juego de adivinanzas en el que el jugador debe adivinar un número generado aleatoriamente.
10        public static void Play()
11        {
12            while (true)
13            {
14                int randno = Newnum(1, 101);
15                int count = 1;
16                while (true)
17                {
18                    Console.Write("Enter a number between 1 and 100(0 to quit):");
19                    int input = Convert.ToInt32(Console.ReadLine());
20                    if (input == 0)
21                        return;
22                    else if (input < randno)
23                    {
24                        Console.WriteLine("Low, try again.");
25                        ++count;
26                        continue;
27                    }
28                    else if (input > randno)
29                    {
30                        Console.WriteLine("High, try again.");
31                    }
32                }
33            }
34        }
35    }
36 }

```

Es una mala práctica de programación no comentar la función que realiza un método. Se propone hacer un pequeño comentario con la función del programa.

1. Dos cosas que no sabía y aprendió en el laboratorio

Aprendí diferentes maneras para poder tener un código más estándar y de mejor calidad para la industria.

2. Una cosa que se le hizo difícil de realizar y explique por qué fue difícil.

Algunos errores que no eran tan obvios.

3. Una cosa que se le hizo fácil de realizar y explique por qué fue fácil.

Algunos errores eran muy obvios de encontrar, entonces fue bastante fácil identificarlos.

4. Indique cuánto tiempo tardó en realizar el laboratorio.

5 horas.