



SOLUCIÓN PROPUESTA

PROGRAMACIÓN ORIENTADA A OBJETOS

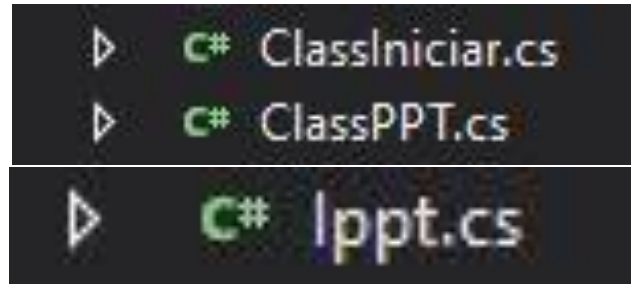
INSTITUTO TECNOLÓGICO DEL SUR DE NAYARIT

MAESTRA: MTI. CINTHIA ANAHÍ MATA BRAVO

EDUARDO MARMOLEJO ORNELAS - 191140006

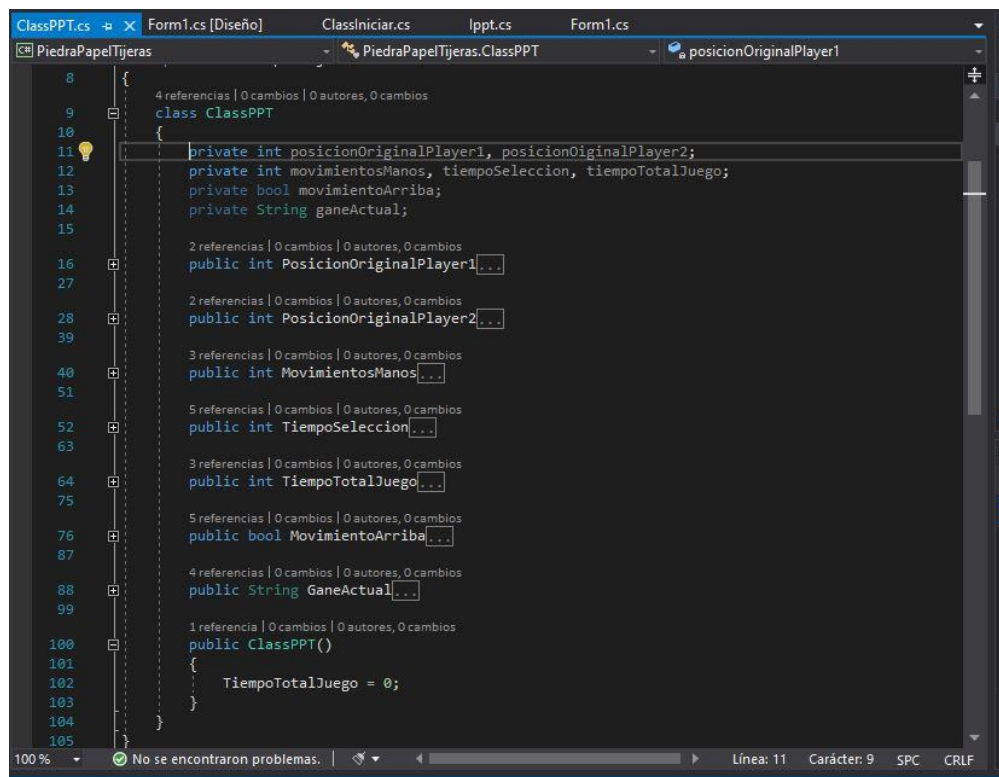
SOLUCIÓN PROPUESTA

Para la construcción de la solución del programa o juego, es que solamente se utilizó dos clases y una interface.



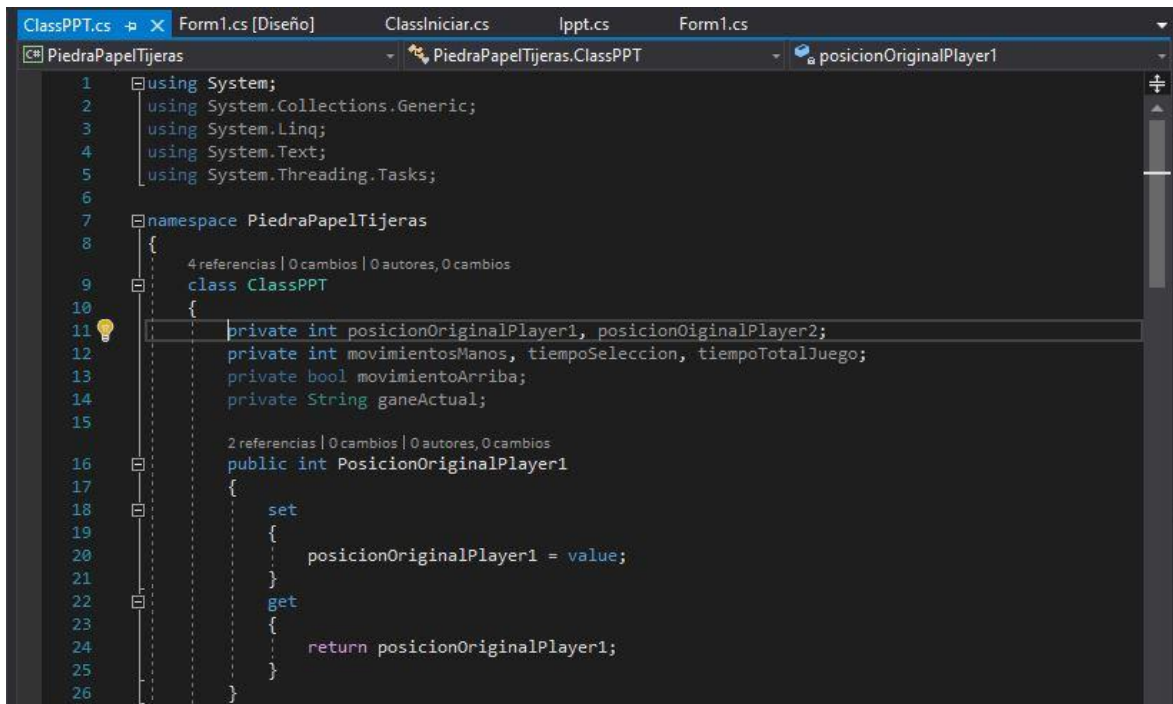
La clase **ClassPPT**, en ella contiene todos los atributos utilizados en privado y también cada atributo tiene su propiedad.

También en la clase padre contiene un constructor la cual solo tiene un único atributo y este atributo es **TiempoTotalJuego** con su inicialización es igual a 0.

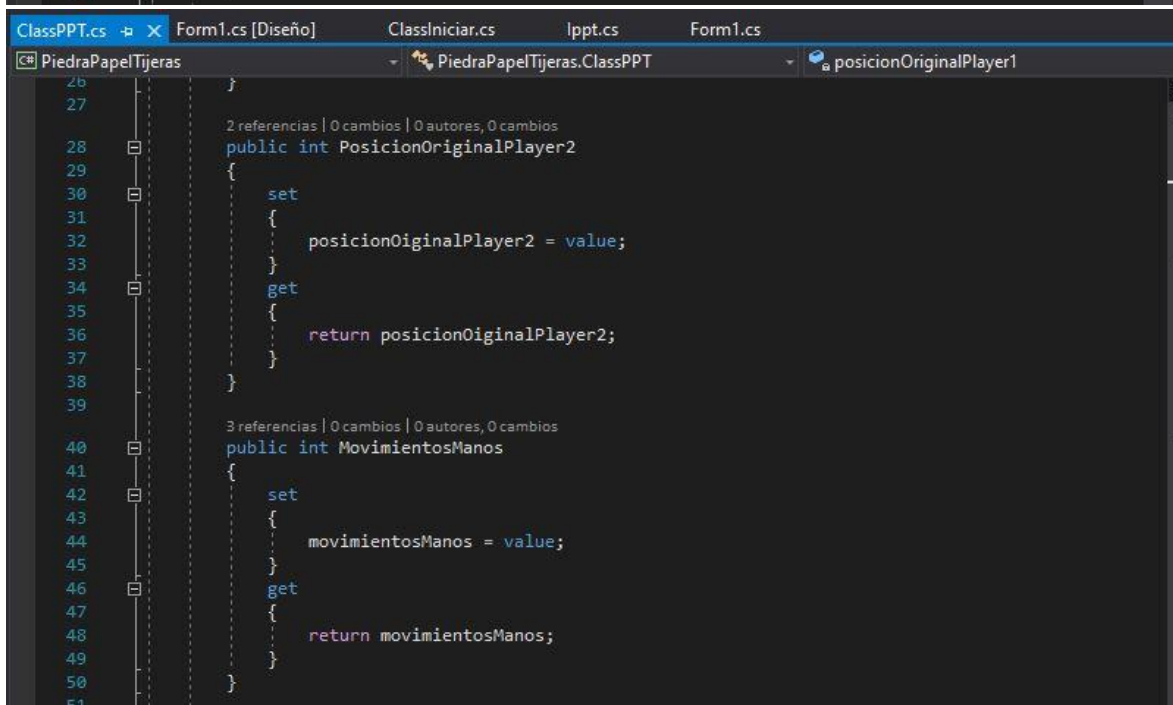
A screenshot of the Visual Studio IDE showing the code for ClassPPT.cs. The code defines a class ClassPPT with several private and public attributes and a constructor. The attributes include posicionOriginalPlayer1, posicionOriginalPlayer2, movimientosManos, tiempoSeleccion, tiempoTotalJuego, movimientoArriba, and ganeActual. The constructor initializes tiempoTotalJuego to 0. The status bar at the bottom indicates "No se encontraron problemas." and "Línea: 11 Carácter: 9 SPC CRLF".

```
8 {
9     4 referencias | 0 cambios | 0 autores, 0 cambios
10     class ClassPPT
11     {
12         private int posicionOriginalPlayer1, posicionOriginalPlayer2;
13         private int movimientosManos, tiempoSeleccion, tiempoTotalJuego;
14         private bool movimientoArriba;
15         private String ganeActual;
16
17         2 referencias | 0 cambios | 0 autores, 0 cambios
18         public int PosicionOriginalPlayer1...
19
20         2 referencias | 0 cambios | 0 autores, 0 cambios
21         public int PosicionOriginalPlayer2...
22
23         3 referencias | 0 cambios | 0 autores, 0 cambios
24         public int MovimientosManos...
25
26         5 referencias | 0 cambios | 0 autores, 0 cambios
27         public int TiempoSeleccion...
28
29         3 referencias | 0 cambios | 0 autores, 0 cambios
30         public int TiempoTotalJuego...
31
32         5 referencias | 0 cambios | 0 autores, 0 cambios
33         public bool MovimientoArriba...
34
35         4 referencias | 0 cambios | 0 autores, 0 cambios
36         public String GaneActual...
37
38         1 referencia | 0 cambios | 0 autores, 0 cambios
39         public ClassPPT()
40         {
41             TiempoTotalJuego = 0;
42         }
43     }
44 }
```

Aquí se muestran capturas de las propiedades de algunos atributos ya mostrados en la captura anterior.

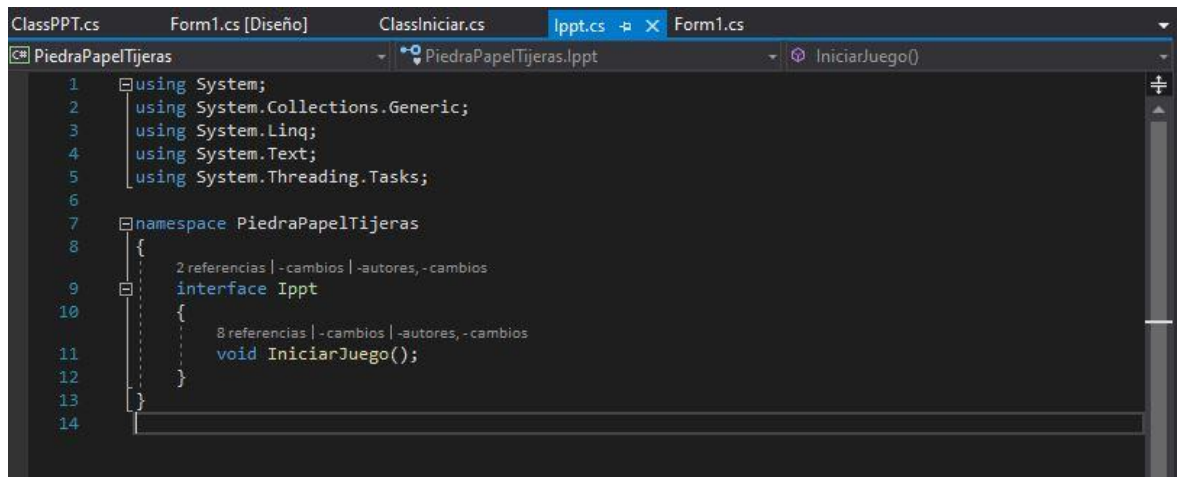


```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace PiedraPapelTijeras
8  {
9      4 referencias | 0 cambios | 0 autores, 0 cambios
10     class ClassPPT
11     {
12         private int posicionOriginalPlayer1, posicionOriginalPlayer2;
13         private int movimientosManos, tiempoSeleccion, tiempoTotalJuego;
14         private bool movimientoArriba;
15         private String ganeActual;
16
17         2 referencias | 0 cambios | 0 autores, 0 cambios
18         public int PosicionOriginalPlayer1
19         {
20             set
21             {
22                 posicionOriginalPlayer1 = value;
23             }
24             get
25             {
26                 return posicionOriginalPlayer1;
27             }
28         }
29     }
```



```
26     }
27
28     2 referencias | 0 cambios | 0 autores, 0 cambios
29     public int PosicionOriginalPlayer2
30     {
31         set
32         {
33             posicionOriginalPlayer2 = value;
34         }
35         get
36         {
37             return posicionOriginalPlayer2;
38         }
39     }
40
41     3 referencias | 0 cambios | 0 autores, 0 cambios
42     public int MovimientosManos
43     {
44         set
45         {
46             movimientosManos = value;
47         }
48         get
49         {
50             return movimientosManos;
51         }
52     }
```

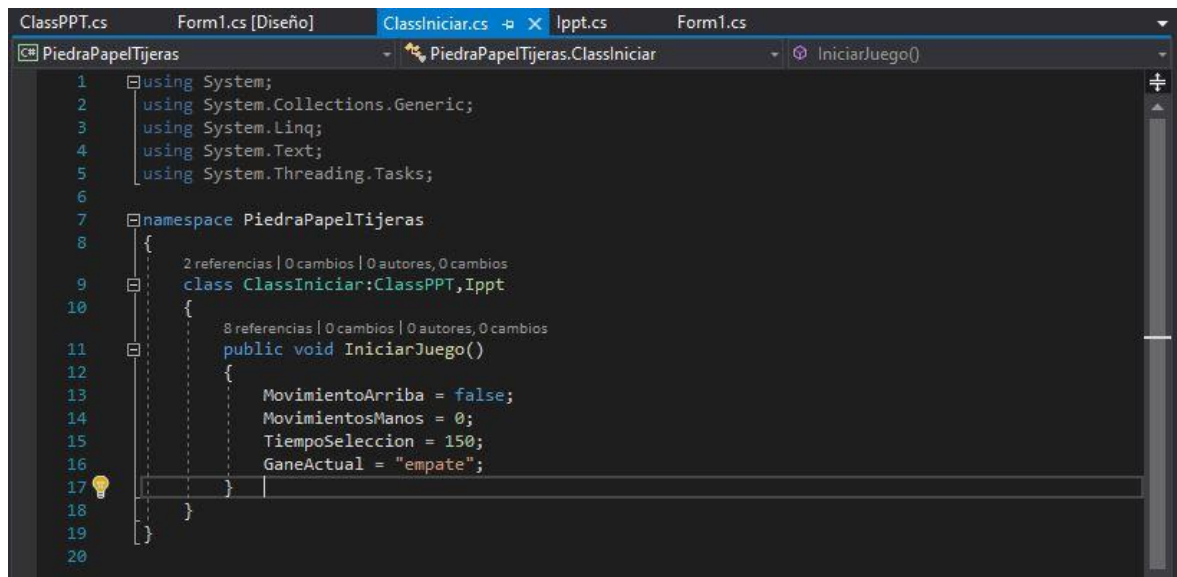
Para la interface, es muy sencilla ya que solo es la definición del método.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace PiedraPapelTijeras
8 {
9     interface Ippt
10     {
11         void IniciarJuego();
12     }
13 }
14
```

Como la interface es asimilado como un contrato, aquí solamente como está definido el método las clases o donde se implementará el método definido deberá de cumplir con estas reglas de la interface.

Para la clase hija, en esta solo es el método implementado con algunos atributos, que solamente se utilizaría en el programa o juego.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace PiedraPapelTijeras
8 {
9     class ClassIniciar:ClassPPT,Ippt
10     {
11         public void IniciarJuego()
12         {
13             MovimientoArriba = false;
14             MovimientosManos = 0;
15             TiempoSeleccion = 150;
16             GaneActual = "empate";
17         }
18     }
19 }
20
```

Aquí en la clase hija es donde se implementa la herencia y polimorfismo, la cuál muestra como es la herencia del padre a la hija y también la implementación de la interface.

```
class ClassIniciar:ClassPPT,Ippt
```

Ahora en la parte de formulario en su interfaz.



Aquí para el juego el jugador deberá seleccionar el botón correspondiente de acuerdo a los colores de fondo, por ejemplo, si las manos ponen tijeras, el usuario debe seleccionar el color amarillo o como dice en la imagen empate, al igual que en las otras opciones, si gana el rojo se deberá de seleccionar el rojo , al igual con el azul, si la opción es correcta, este deberá sumarle un punto y hacer sonar un sonido de puntos, de lo contrario un de error y no se le sumarán puntos.

El 0 que se muestra a la derecha en la imagen, en este mostrará el tiempo que tendrá en seleccionar el usuario. Llegando a un límite de tiempo saldrá un cuadro de juego terminado, mostrándole los puntos que obtuvo.



Ahora en la parte de la codificación también se implementó la herencia y polimorfismo en la interface con el formulario, ya que aquí también se implementó el método.

```
public partial class Form1 : Form, Ippt
```

Aquí la implementación del método

```
//Inicialización del método implementado en el formulario con los timer y el objeto
//del método
8 referencias | 0 cambios | 0 autores, 0 cambios
public void IniciarJuego() {
    timer1.Enabled = true;
    timer2.Enabled = true;
    timer1.Start();
    timer2.Start();
    objIniciar.IniciarJuego();
}
```

Con el obj de abajo es donde se le hace la llamada al método.

```
objIniciar.IniciarJuego();
```