

Materia: Programación Orientada a Objetos

Nombre: Miramontes Luis Uriel

27/03/2020

Documentación

JuegoUnidad4 – “Memorama

con temática de JoJo’s Bizarre Adventure”

Maestra: Ing. Cinthia Anahí Mata Bravo



Descripción del juego

El memorama es un juego que tiene como objetivo desarrollar y agilizar la capacidad de poder como su propio nombre lo dice, memorizar objetos. Así que se creo este juego con el fin de lograr este objetivo.

Para este juego se usó la temática de los personajes del anime llamado “JoJo’s Bizarre Adventure - Golden Wind” donde el usuario deberá obtener el menor número de movimientos posibles durante el juego y a la vez entretenerse y agilizar su memoria.

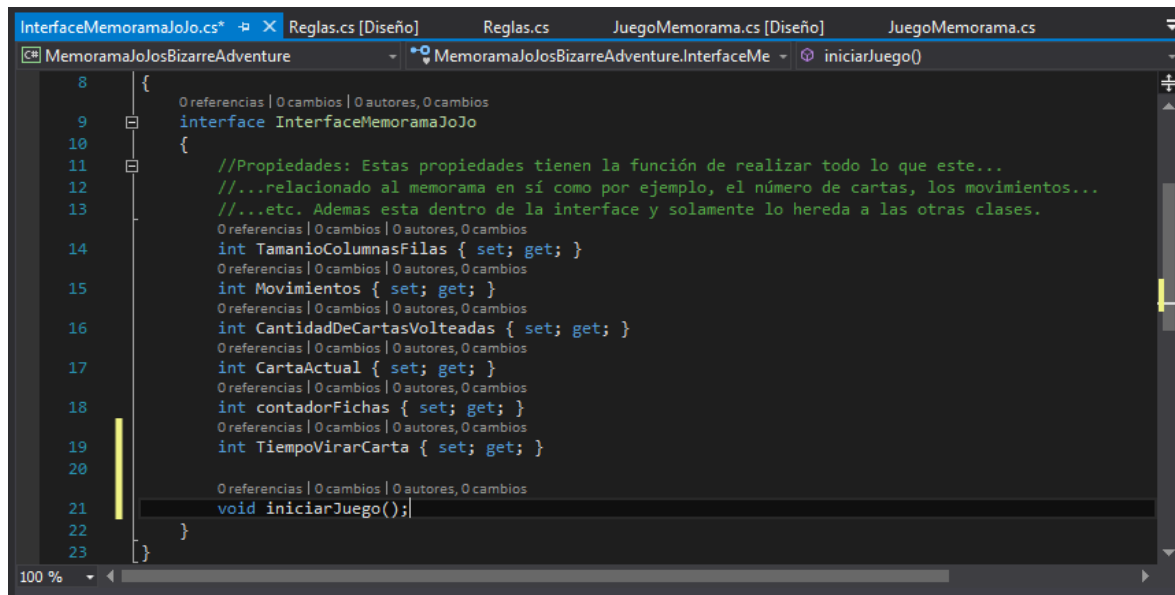
Para saber si el jugador ha ganado el juego, saldrá un mensaje diciendo “Eccelente ¡Di Molto!” indicando el fin de la partida :D

Personajes del juego:



Solución propuesta

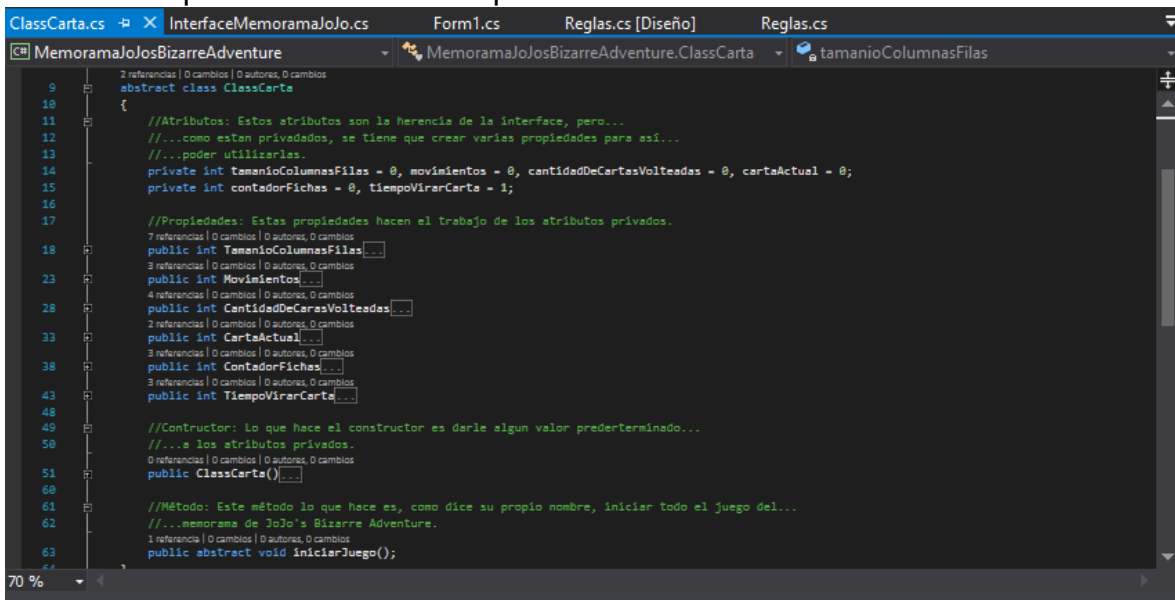
Paso 1. Para poder programar el memorama primero se debe de crear una interface que en este caso se llama “InterfaceMemoramaJoJo”.



```
8 {
9     0 referencias | 0 cambios | 0 autores, 0 cambios
10     interface InterfaceMemoramaJoJo
11     {
12         //Propiedades: Estas propiedades tienen la función de realizar todo lo que este...
13         //...relacionado al memorama en sí como por ejemplo, el número de cartas, los movimientos...
14         //...etc. Ademas esta dentro de la interface y solamente lo hereda a las otras clases.
15         0 referencias | 0 cambios | 0 autores, 0 cambios
16         int TamanoColumnasFilas { set; get; }
17         0 referencias | 0 cambios | 0 autores, 0 cambios
18         int Movimientos { set; get; }
19         0 referencias | 0 cambios | 0 autores, 0 cambios
20         int CantidadDeCartasVolteadas { set; get; }
21         0 referencias | 0 cambios | 0 autores, 0 cambios
22         int CartaActual { set; get; }
23         0 referencias | 0 cambios | 0 autores, 0 cambios
24         int contadorFichas { set; get; }
25         0 referencias | 0 cambios | 0 autores, 0 cambios
26         int TiempoVirarCarta { set; get; }
27
28         0 referencias | 0 cambios | 0 autores, 0 cambios
29         void iniciarJuego();
30     }
31 }
```

Donde se crean las propiedades de tipo entero “int” para después heredarlas a las clases siguientes, estas propiedades sirven para poder empezar a formar las cartas del memorama ya sea el número de cartas, los movimientos, entre otros...

Paso 2. Después se crea la clase padre llamada “ClassCarta”.



```
9 2 referencias | 0 cambios | 0 autores, 0 cambios
10 abstract class ClassCarta
11 {
12     //Atributos: Estos atributos son la herencia de la interface, pero...
13     //...como estan privados, se tiene que crear varias propiedades para así...
14     //...poder utilizarlos.
15     private int tamanoColumnasFilas = 0, movimientos = 0, cantidadDeCartasVolteadas = 0, cartaActual = 0;
16     private int contadorFichas = 0, tiempoVirarCarta = 1;
17
18     //Propiedades: Estas propiedades hacen el trabajo de los atributos privados.
19     7 referencias | 0 cambios | 0 autores, 0 cambios
20     public int TamanoColumnasFilas { get; set; }
21     3 referencias | 0 cambios | 0 autores, 0 cambios
22     public int Movimientos { get; set; }
23     4 referencias | 0 cambios | 0 autores, 0 cambios
24     public int CantidadDeCartasVolteadas { get; set; }
25     2 referencias | 0 cambios | 0 autores, 0 cambios
26     public int CartaActual { get; set; }
27     3 referencias | 0 cambios | 0 autores, 0 cambios
28     public int ContadorFichas { get; set; }
29     3 referencias | 0 cambios | 0 autores, 0 cambios
30     public int TiempoVirarCarta { get; set; }
31
32     //Constructor: Lo que hace el constructor es darle algun valor predefinido...
33     //...a los atributos privados.
34     0 referencias | 0 cambios | 0 autores, 0 cambios
35     public ClassCarta() { }
36
37     //Método: Este método lo que hace es, como dice su propio nombre, iniciar todo el juego del...
38     //...memorama de JoJo's Bizarre Adventure.
39     1 referencias | 0 cambios | 0 autores, 0 cambios
40     public abstract void iniciarJuego();
41 }
```

Donde las propiedades de la interface se heredan y como es abstracta solamente almacena los datos creados de la interface que servirán para las clases hijas.

Paso 3. Se crea la clase hija que en este caso se llama “Memorama”.

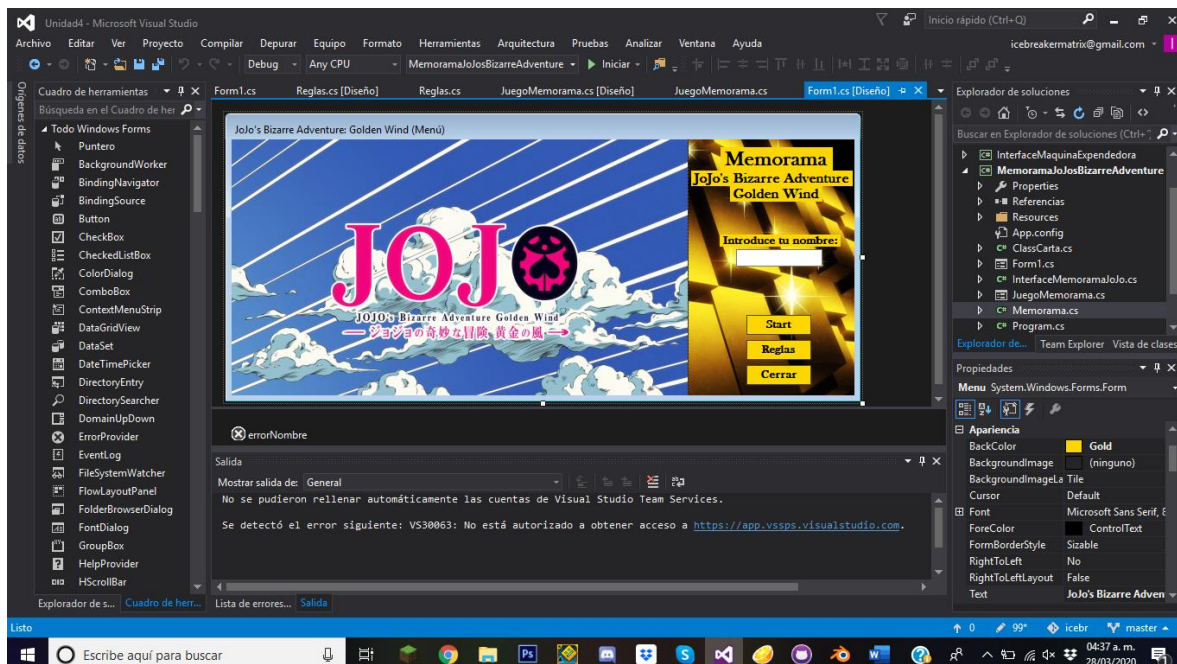
```

Memorama.cs  InterfaceMemoramaJoJo.cs  Form1.cs  Reglas.cs [Diseño]  Reglas.cs
MemoramaJoJosBizarreAdventure
20
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace MemoramaJoJosBizarreAdventure
8  {
9      class Memorama:ClassCarta
10     {
11         //Método: Este método fue heredado de la clase padre llamada "ClassCarta", que ya antes...
12         //...mencionado, este método lo que hace es, como dice su propio nombre, iniciar todo el...
13         //...juego del memorama de JoJo's Bizarre Adventure, aunque aqui mediante el "override"...
14         //...lo puedes sobrescribir ya que es heredado.
15         1 referencia | 0 cambios | 0 autores, 0 cambios
16         public override void iniciarJuego()
17         {
18         }
19     }
20
100 %

```

Donde se hereda el método de la clase padre, donde además se puede sobrescribir con el “override” lo que se necesite complementar al código.

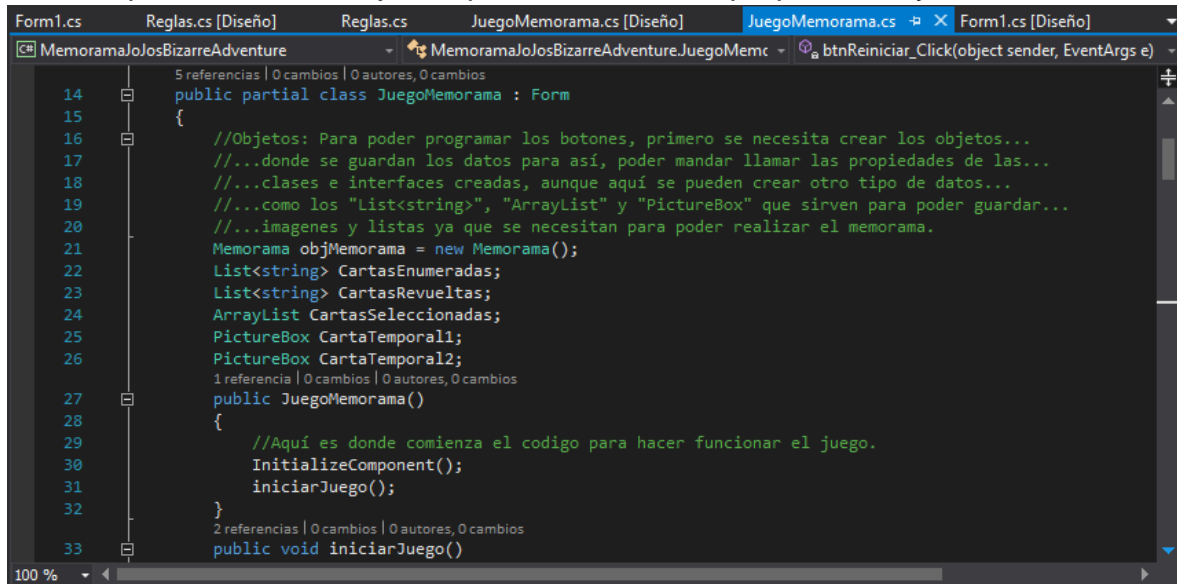
Paso 4. Diseñar las formas.



Por medio del diseño de las formas se insertarán imágenes, labels y botones que servirán para poder empezar a edificar el código, ya que por medio de los botones se harán las acciones del memorama.

Paso 5. Programar los botones.

En esta parte se crean objetos que llaman a las propiedades y métodos creadas



```
Form1.cs Reglas.cs [Diseño] Reglas.cs JuegoMemorama.cs [Diseño] JuegoMemorama.cs x Form1.cs [Diseño]
MemoramaJoJo'sBizarreAdventure
5 referencias | 0 cambios | 0 autores, 0 cambios
14 public partial class JuegoMemorama : Form
15 {
16     //Objetos: Para poder programar los botones, primero se necesita crear los objetos...
17     //...donde se guardan los datos para así, poder mandar llamar las propiedades de las...
18     //...clases e interfaces creadas, aunque aquí se pueden crear otro tipo de datos...
19     //...como los "List<string>", "ArrayList" y "PictureBox" que sirven para poder guardar...
20     //...imagenes y listas ya que se necesitan para poder realizar el memorama.
21     Memorama objMemorama = new Memorama();
22     List<string> CartasEnumeradas;
23     List<string> CartasRevueltas;
24     ArrayList CartasSeleccionadas;
25     PictureBox CartaTemporal1;
26     PictureBox CartaTemporal2;
27     public JuegoMemorama()
28     {
29         //Aquí es donde comienza el código para hacer funcionar el juego.
30         InitializeComponent();
31         iniciarJuego();
32     }
33     2 referencias | 0 cambios | 0 autores, 0 cambios
34     public void iniciarJuego()
```

mediante las clases e interfaces para así estructurar finalmente el código.

Instrucciones del juego

1. Al iniciar el juego te abrirá un menú con el logo de “JoJo’s Bizarre Adventure – Golden Wind” donde habrá un textbox y 3 botones:
 - Start: Este botón abrirá el juego, pero necesitaras ingresar primero tu nombre de jugador en el textbox si no, no te dejara abrir el juego
 - Reglas: Aquí se abrirá una ventana donde te aparecerá las reglas del juego para que las puedas leer, si le das click al botón “OK” se cerrara.
 - Cerrar: Solamente cierra el menú.



2. Al momento de iniciar a jugar aparecerá tu nombre ingresado en un label, tu record de movimientos en cero, el memorama por medio de un “PanelBox” y 2 botones:
- Reiniciar: Este botón hará que reinicie tu partida incluyendo las cartas (las volverá a revolver) y tu record de movimientos.
 - Cerrar: Solamente cierra el juego.



3. Al momento de que des click a una carta, tu record de movimientos incrementara y si encuentras algún par, las marcara con la imagen del personaje.



-

```

classDiagram
    class InterfaceMemoramaJoJo {
        +iniciarJuego()
    }
    class ClasePadreClassCarta {
        -TamanoColumnasFilas
        -Movimientos
        -CantidadDeCartasVolteadas
        -CartaActual
        -ContadorFichas
        -TiempoVirarCarta
        +iniciarJuego()
    }
    class ClaseHijaMemorama {
        +iniciarJuego()
    }
    InterfaceMemoramaJoJo --> ClasePadreClassCarta
    ClasePadreClassCarta --> ClaseHijaMemorama

```