

Participantes:

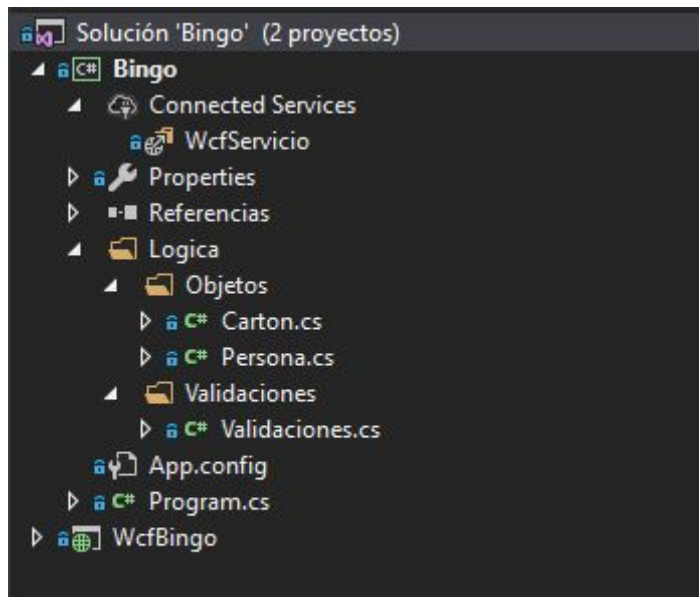
José Andrés Zúñiga Martínez - 20160110228

Fabián Andrés Rodríguez Meza - 2015120121

Alonso José Mesén Quesada - 20160110243

Aaron Vargas Malavasi - 20160110550

se dividió el código por carpetas con la finalidad de un mejor entendimiento de la aplicación.



Se decidió utilizar dos objetos uno cartón y otro persona con la finalidad de hacer una multilista de jugadores y tener un mejor rendimiento a la hora de ejecución de la aplicación.



```
Bingo Bingo.Carton
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Collections;
7  namespace Bingo
8  {
9      class Carton
10     {
11         private bool haGanado { get; set; }
12         private object [,] carton { get; set; }
13     }
14
15     public Carton(bool haGanado, object[,] carton) ...
16
17     public object[,] getCartonActual() ...
18
19     public void Imprimir() ...
20
21 }
22 }
```

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading;
7  using System.Threading.Tasks;
8
9  namespace Bingo
10 {
11     class Persona
12     {
13         private string Name { get; set; }
14         private List<Carton> _cartones = new List<Carton>();
15
16         public IList<Carton> cartones { get { return _cartones; } }
17
18         public Persona(string name, List<Carton> cartones) ...
19
20         public void GetCrearCs(int cant) ...
21
22         public ArrayList Llenado(ArrayList Numeros, object[,] Cart) ...
23
24         public object[,] FormarCarton(object[,] Cart) ...
25
26         public object[,] LlenarCarton(object[,] Cart, ArrayList Numeros, int B, int I, int N, int G, int O) ...
27
28         public void Imprimir() ...
29
30         public string NombreUsuario() ...
31
32         public object getCarton(int i) ...
33     }
34 }
```

Dentro de la carpeta validaciones se encuentra una clase que se encarga de validar el modo de juego seleccionado y de las validaciones varias que debe tener el juego para poder disfrutar de la mejor manera, cumpliendo con el requerimiento que en caso de necesitar crear otro modo de juego se afecte lo menos posible lo demás.



```
8 namespace Bingo
9 {
10     class Validaciones
11     {
12         static bool Ganador = false;
13
14         public static void Revision(int Tipo) ...
15
16         public static void LlenarRevision(int tipo, Persona y, int posicion) ...
17
18         public static bool YaGano(ArrayList numerosTiene) ...
19
20         public static void ImprimirCarton(object[,] carton, ArrayList numerosTiene) ...
21
22         public static bool RevisarBingo(ArrayList n, int x) ...
23
24         public static ArrayList CuatroEsquina(ArrayList numerosTiene, object[,] carton) ...
25
26         public static ArrayList LetraH(ArrayList numerosTiene, object[,] carton) ...
27
28         public static ArrayList LetraX(ArrayList numerosTiene, object[,] carton) ...
29
30         public static ArrayList LetraO(ArrayList numerosTiene, object[,] carton) ...
31
32         public static ArrayList CartonLleno(ArrayList numerosTiene, object[,] carton) ...
33
34     }
35 }
```

En la clase principal podemos observar el menú de ejecución el cual nos va a brindar una guía de cuales son las acciones que se pueden realizar dentro del programa.

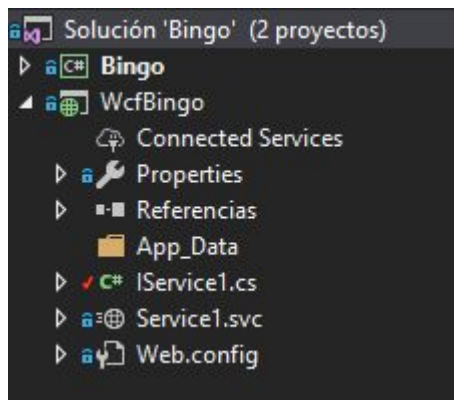
```
public static void TextoMenu()
{
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("Modos de juego");
    Console.WriteLine("1.Juego 4 Esquinas");
    Console.WriteLine("2.Juego H");
    Console.WriteLine("3.Juego X");
    Console.WriteLine("4.Juego O");
    Console.WriteLine("5.Juego Clasico (Carton Lleno)");
    Console.WriteLine("Introduzca un Numero:");
}
```

```
public static void Menu()
{
    int num = 1;
    using (WcfServicio.WCFBingoClient WCFInstancia = new WcfServicio.WCFBingoClient())
    {
        var ejemplo = WCFInstancia.MENU(num);
        var mensaje = ejemplo.mensaje;
        Console.WriteLine(mensaje);
        Console.ReadLine();
    }

    Jugadores();
    int Tipo = 0;
    bool esNumero = true;
    string TipoJuego = " ";
    do
    {
        TextoMenu();
        Console.WriteLine("Seleccione el modo de juego:");
        TipoJuego = Console.ReadLine();
        esNumero = int.TryParse(TipoJuego, out Tipo);
    } while (!esNumero);
    Tipo = System.Convert.ToInt32(TipoJuego);

    for (int i = 0; i <= 70; i++)
    {
        NumeroSale();
        Validaciones.Revision(Tipo);
        if (EsGanador == true)
        {
            Console.ReadLine();
            return;
        }
    }
    Console.ReadLine();
}
```

Por último está el WCF, el cual se generó en un proyecto aparte dentro de la misma solución.



```
10 namespace WcfBingo
11 {
12     // NOTA: puede usar el comando "Rename" del menú "Refactorizar" para cambiar el nombre de interfaz "IService1" en el código y en el archivo
13     [ServiceContract]
14     public interface IWcfBingo
15     {
16
17         [OperationContract]
18         Menu MENU(int opcion);
19
20         // TODO: agregue aquí sus operaciones de servicio
21     }
22
23     // Utilice un contrato de datos, como se ilustra en el ejemplo siguiente, para agregar tipos compuestos a las operaciones de servicio.
24     [DataContract]
25     public class Menu
26     {
27         [DataMember]
28         public string mensaje { get; set; }
29     }
30 }
31
32
```

```
10 namespace WcfBingo
11 {
12     // NOTA: puede usar el comando "Rename" del menú "Refactorizar" para cambiar el nombre de clase "Service1" en el código, en svc y en
13     // NOTE: para iniciar el Cliente de prueba WCF para probar este servicio, seleccione Service1.svc o Service1.svc.cs en el Explorador
14     public class WcfBingo : IWcfBingo
15     {
16
17         public Menu MENU(int opcion)
18         {
19
20             if(opcion == 1)
21             {
22
23                 return new Menu() { mensaje = "Bienvenido al Bingo \n\n" +
24                                         "Instrucciones: \n" +
25                                         "1) Debe elegir la cantidad de jugadores por partida \n" +
26                                         "2) Debe indicar el nombre de cada jugador \n" +
27                                         "3) Debe indicar la cantidad de cartones por jugador \n" +
28                                         "4) Debe seleccionar el modo de juego \n\n"
29                                     };
30             }
31             else
32             {
33                 throw new Exception("Error");
34             }
35
36             throw new Exception("Error");
37         }
38     }
39 }
40
41
42
```

Ahora vamos a ver el proyecto en ejecución:

Lo primero que vamos a ver es una pequeña descripción del programa una guía que es generada desde el WCF.

```
Bienvenido al Bingo

Instrucciones:
1> Debe elegir la cantidad de jugadores por partida
2> Debe indicar el nombre de cada jugador
3> Debe indicar la cantidad de cartones por jugador
4> Debe seleccionar el modo de juego
```

Luego de esto el usuario debe indicar la cantidad de jugadores presentes en el juego:

```
Digite la cantidad de usuarios:
2
```

Se deben escoger los nombres de cada usuario e inmediatamente la cantidad de cartones que cada usuario desea utilizar en el juego:

```
Digite el Nombre de Usuario:
Alonso
Digite la cantidad de cartones que desea:Alonso
2
```

```
Digite el Nombre de Usuario:
Fabiam
Digite la cantidad de cartones que desea:Fabiam
1
```

B	I	N	G	O
14	22	39	56	61
0	29	37	54	71
1	25		52	63
10	24	40	57	73
8	18	31	58	70

B	I	N	G	O
12	25	44	54	63
0	27	40	59	69
3	17		49	68
4	19	39	53	65
8	28	36	50	73
B	I	N	G	O
6	22	42	55	66
0	21	37	57	70
3	20		52	73
14	28	44	48	69
7	25	38	53	71

```

Modos de juego
1.Juego 4 Esquinas
2.Juego H
3.Juego X
4.Juego O
5.Juego Clasico <Carton LLeno>
Introduzca un Numero:
Seleccione el modo de juego:

```

Luego de esto el programa va a empezar a sacar número por número de manera aleatoria y a la vez también válida cuales cartones y de cuáles usuarios tienen el número que salió, y a su vez informa si hay, o no ganador.

Fabiam				
B	I	N	G	O
14	22	39	56	61
0	29	37	54	71
1	25		52	63
10	24	40	57	73
8	18	31	58	70
No hubo ganadores				
En la columna de la G, Numero: 48				
Alonso				
B	I	N	G	O
12	25	44	54	63
0	27	40	59	69
3	17		49	68
4	19	39	53	65
8	28	36	50	73

En caso de haber ganador el programa informa cual usuario y con cual cartón a ganado el juego y finaliza el programa.

```
Ya tenemos ganador: Alonso
Con el carton

B      I      N      G      O
6      22     42     55     66
0      21     37     57     70
3      20           52     73
14     28     44     48     69
7      25     38     53     71
```