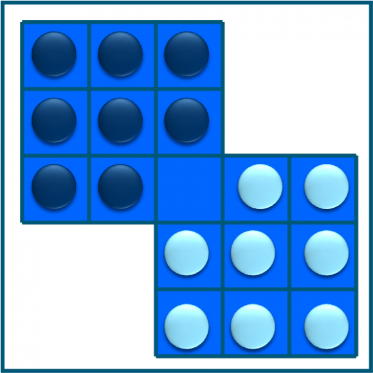
Laura Rojas Solís

Lvrs0505@gmai.com

07/07/2021

Version 01.03



SOLUTION

CHANGE OFCHIPS

**CONTROL SHEET**

|  |  |  |
| --- | --- | --- |
| **item** | **State %** | **remarks** |
| **Sections of the game** |  |  |
|  Home screen | Complete (100%) |  |
|  Game solution | Incomplete |  |
|  Archives | Complete (100%) |  |
|  Reports | Incomplete | reports are missing. |

# **INDEX**

[**INDEX** 2](#_Toc76432180)

[**1** **PRINCIPAL WINDOW** 3](#_Toc76432181)

[**1.1** **Your full name (all team members are placed).** 3](#_Toc76432182)

[**1.2** **A photograph of you (must be inside the project and all of the team members are placed).** 4](#_Toc76432183)

[**1.3** **The current date and time of the system.** 4](#_Toc76432184)

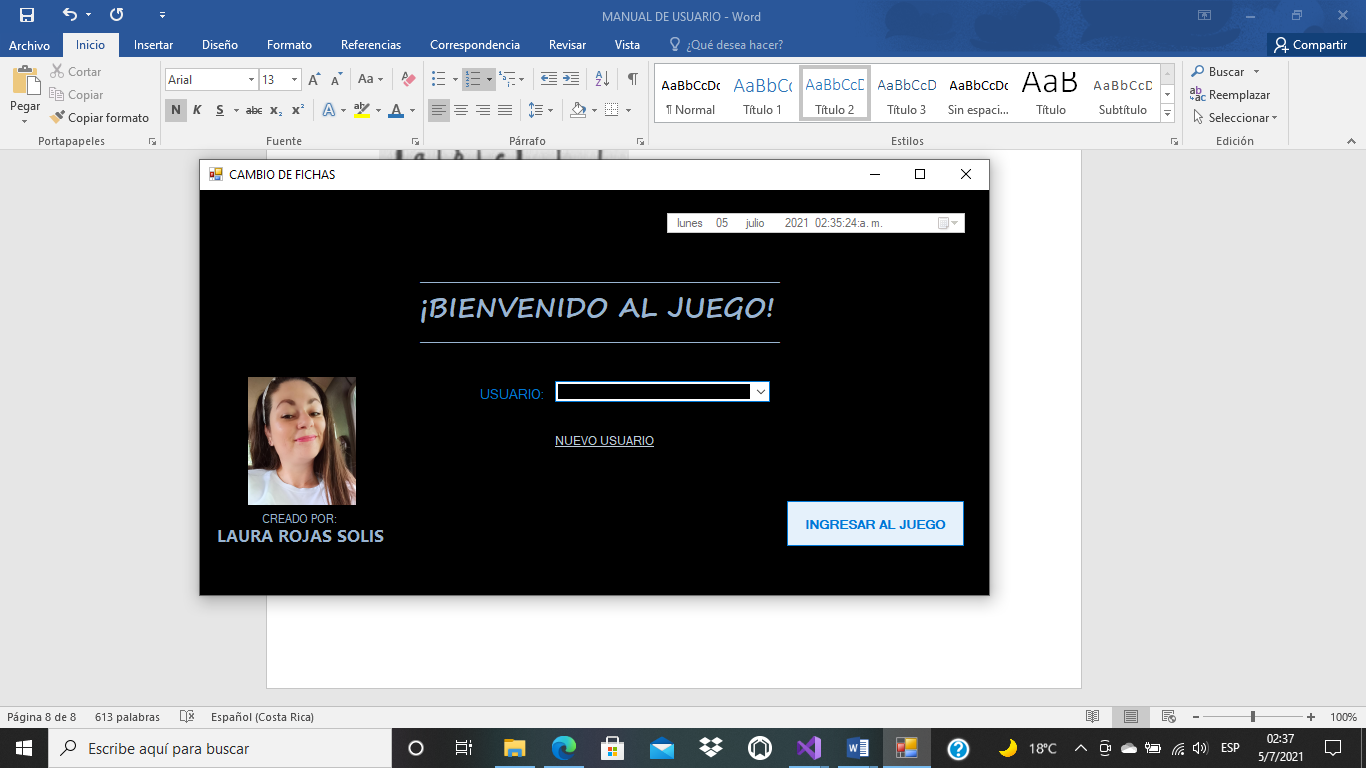
[**1.4** **A welcome message to the game, in addition the user is asked to enter the name (it will be assumed that repeated names will not be entered).** 5](#_Toc76432185)

[**1.5** **Introduce a new user:** 6](#_Toc76432186)

[**1.6** **A button that allows access to the second system window.** 10](#_Toc76432187)

[**2** **DATA BASE** 12](#_Toc76432188)

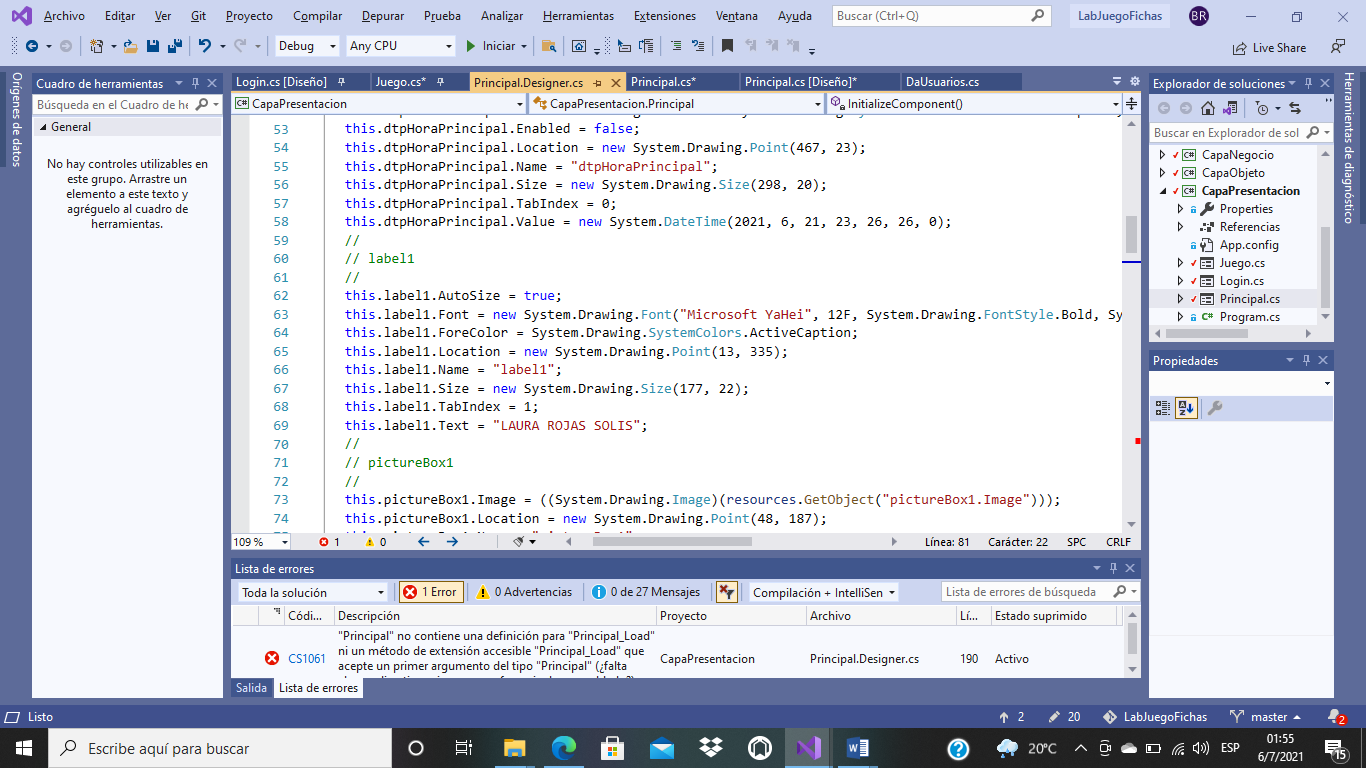
# **PRINCIPAL WINDOW**



There should be a home screen, where the following items should appear:

## **Your full name (all team members are placed).**

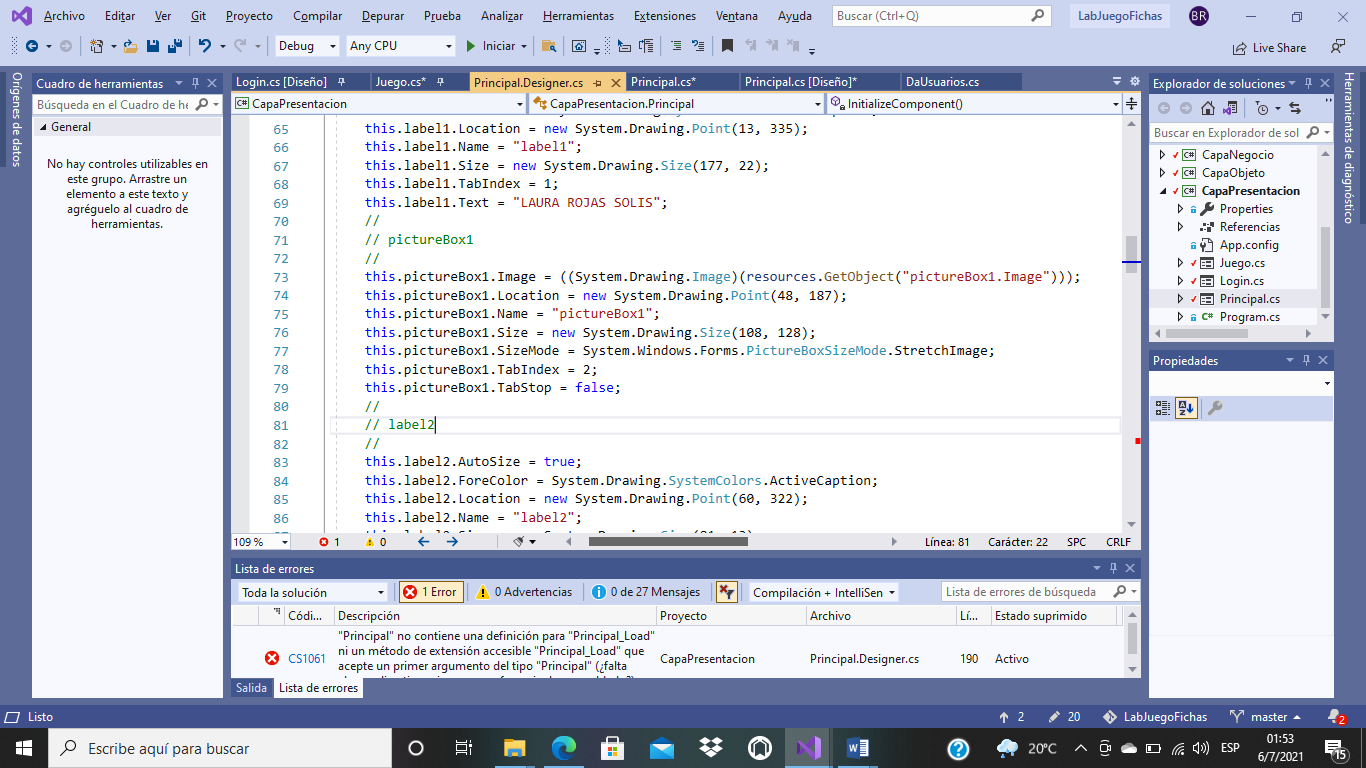
Create a label, in this properties label do you can write: Text “EVERY NAME DO YOU WANT”.



## **A photograph of you (must be inside the project and all of the team members are placed).**

Do select a pictureBox on Visual Studio tools, and do you select your potography and them, put this in your Solution Folder.

Chose your potography



## **The current date and time of the system.**

Chose a datepickerTime , and them give it a Custom Format:

“dddd” to write name of day

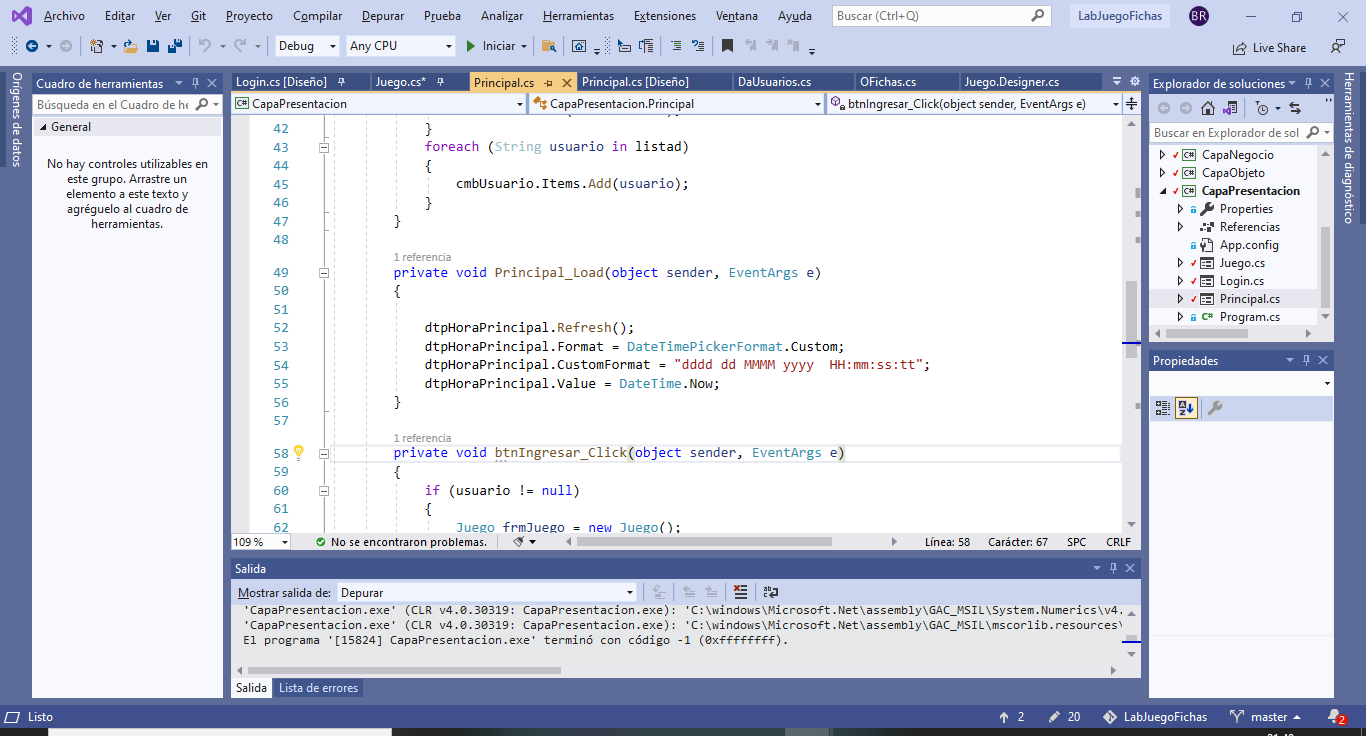
“dd” the number of day

“MMMM” to appear name of the month in letters

“yyyy” number of the actual year.

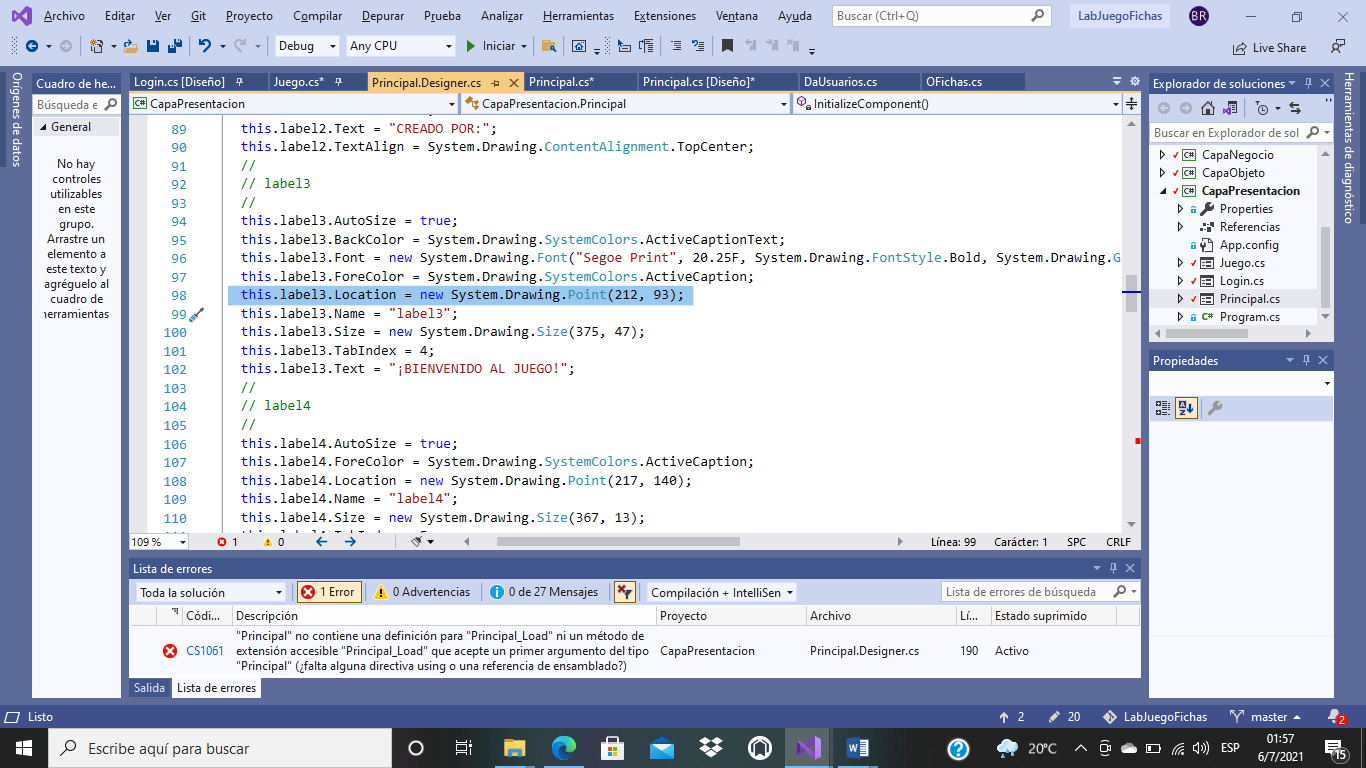
“HH:mm:ss:tt” to appear the long hour

Also do you need write datepickerTime**. Value**= DateTime.Now to actualice at System hour correct.



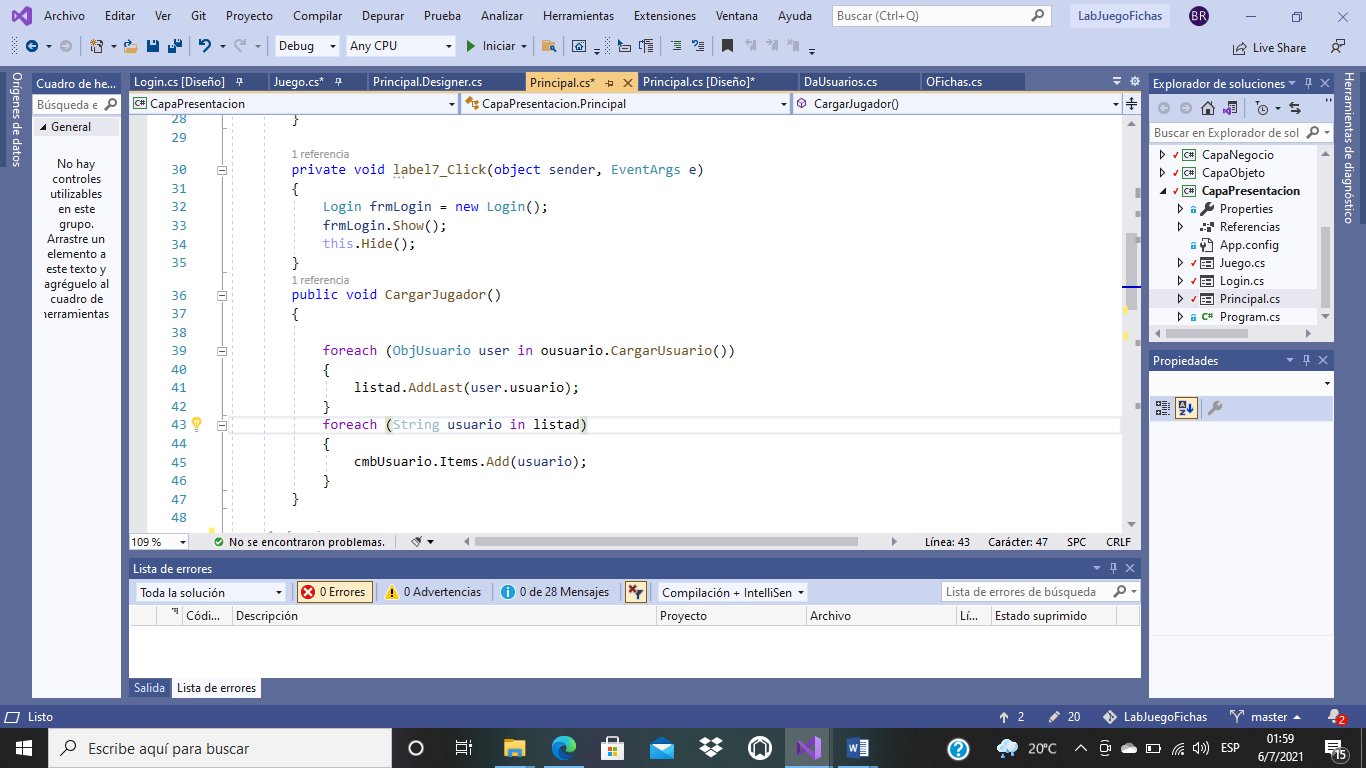
## **A welcome message to the game, in addition the user is asked to enter the name (it will be assumed that repeated names will not be entered).**

**WELCOME MESSAGE:**  in a Label do you can write a Welcome Message for the user

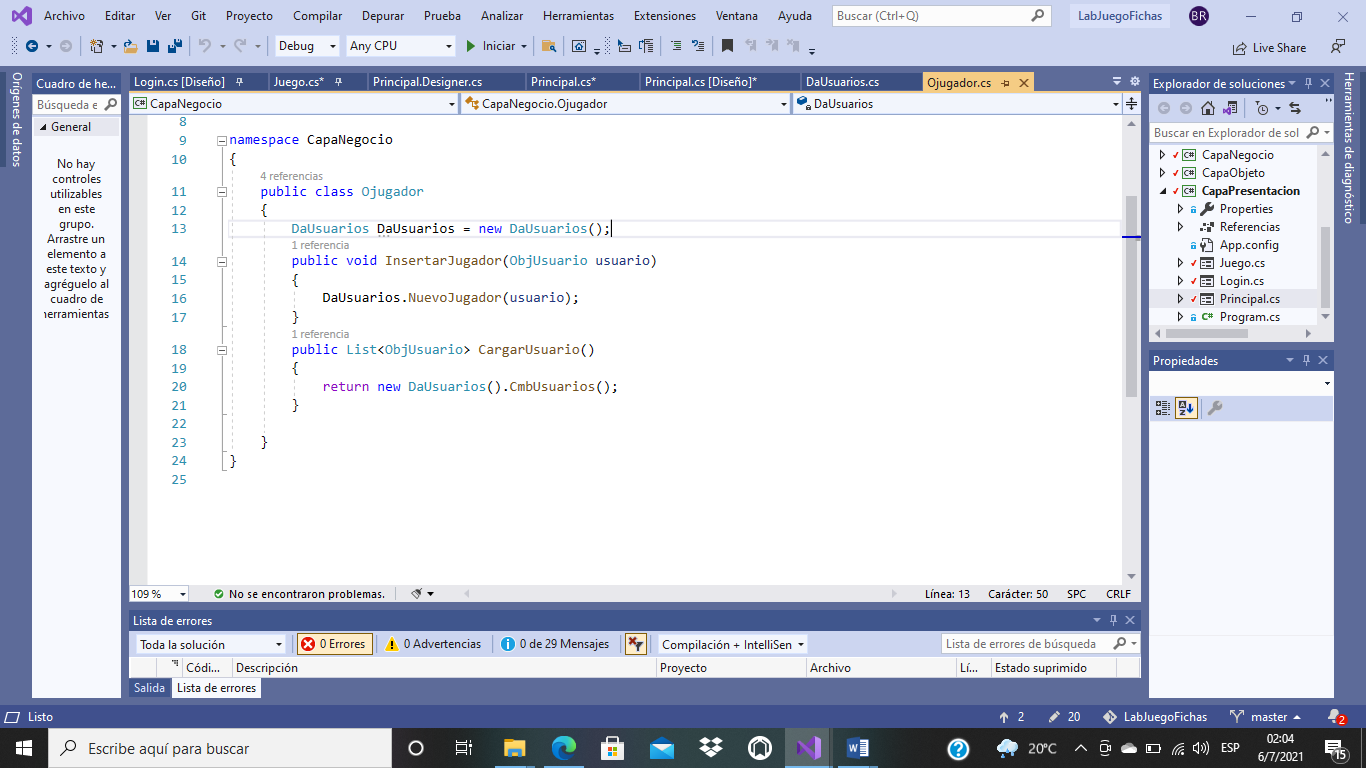


**USER:** I preffer obtain users data from the DataBase and show the user alternatives on a ComboBox

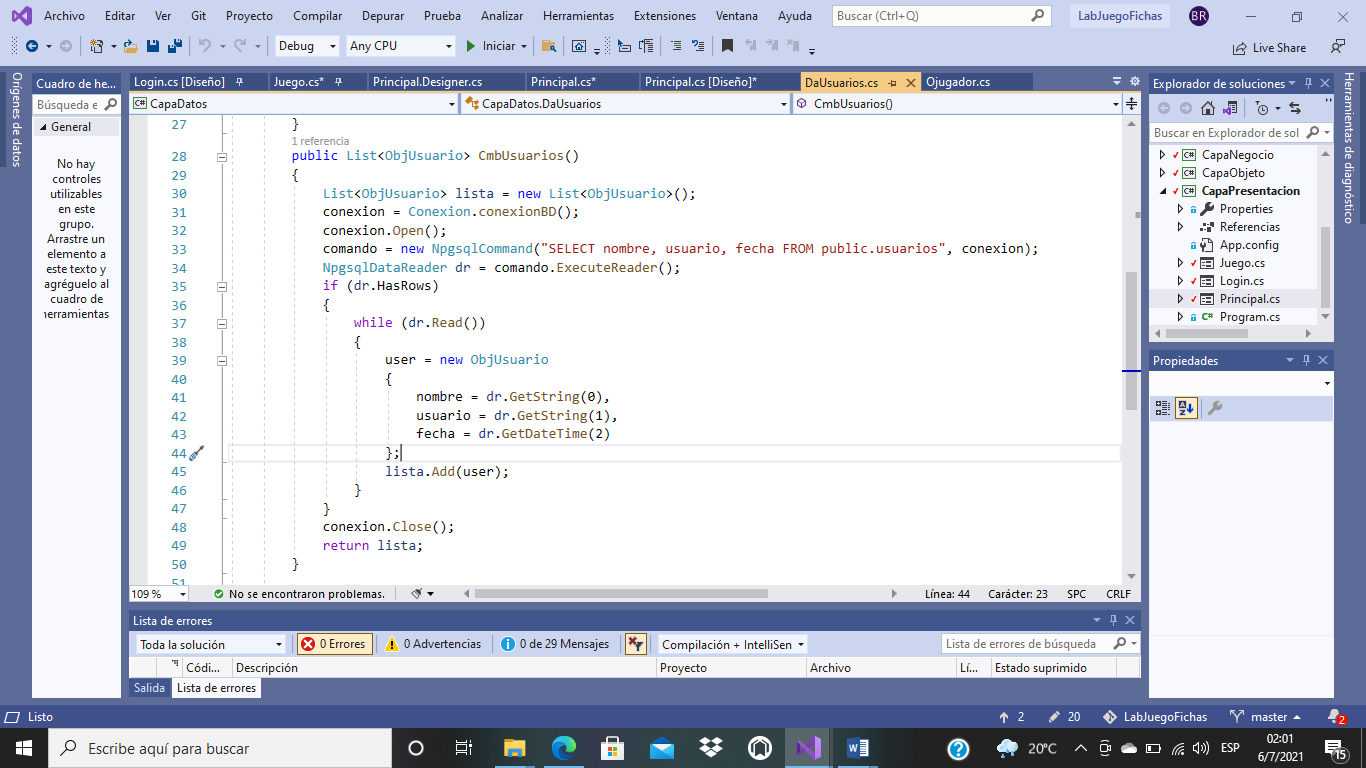
(presentation cape) Through a object, obtain every register user, and show this in a comboBox



(business cape) Receive the user Object and call load information from the table of the DataBase

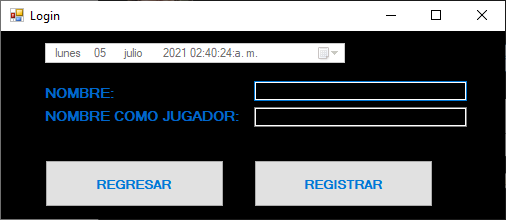


(data cape ): in this cape look for the user information on the user table from DataBase.



## **Introduce a new user:**

Do you need think in the possibility that the user is new name and needs to be registered, to avoid duplicate data, in the database choose a user as a primary Key-



(presentation cap): create basic methods to register the new user.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using CapaObjeto;

using CapaNegocio;

namespace CapaPresentacion

{

public partial class Login : Form

{

ObjUsuario user = new ObjUsuario();

Ojugador ojugador = new Ojugador();

public Login()

{

InitializeComponent();

}

private void btnUsuario\_Click(object sender, EventArgs e)

{

obtenerdatos();

ojugador.InsertarJugador(user);

System.Windows.Forms.MessageBox.Show("USUARIO AGREGADO CON EXITO ");

Principal();

}

public void obtenerdatos()

{

user.nombre = txtNombreUsuario.Text;

user.usuario = txtJugador.Text;

user.fecha = dtpFecha.Value;

}

private void btnRegresar\_Click(object sender, EventArgs e)

{

Principal();

}

public void Principal()

{

Principal frmPrincipal = new Principal();

frmPrincipal.Show();

this.Hide();

}

private void Login\_Load(object sender, EventArgs e)

{

dtpFecha.Refresh();

dtpFecha.Format = DateTimePickerFormat.Custom;

dtpFecha.CustomFormat = "dddd dd MMMM yyyy HH:mm:ss:tt";

dtpFecha.Value = DateTime.Now;

}

}

}

(Object cape): in this cap, remember we have an information as a user’s table from database

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using CapaDatos;

using CapaObjeto;

namespace CapaNegocio

{

public class Ojugador

{

DaUsuarios DaUsuarios = new DaUsuarios();

public void InsertarJugador(ObjUsuario usuario)

{

DaUsuarios.NuevoJugador(usuario);

}

public List<ObjUsuario> CargarUsuario()

{

return new DaUsuarios().CmbUsuarios();

}

}

}

(data cape): will need conect the user´s table from database to load information on the system.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Npgsql;

using CapaObjeto;

namespace CapaDatos

{

public class DaUsuarios

{

public NpgsqlCommand comando;

public NpgsqlConnection conexion;

Conexion Daconexion = new Conexion();

ObjUsuario user = new ObjUsuario();

public void NuevoJugador(ObjUsuario jugador)

{

conexion = Conexion.conexionBD();

conexion.Open();

comando = new NpgsqlCommand("INSERT INTO usuarios (nombre, usuario, fecha)" +

"VALUES ('" + jugador.nombre + "','" + jugador.usuario + "','" + jugador.fecha + "')", conexion);

comando.ExecuteNonQuery();

conexion.Close();

}

public List<ObjUsuario> CmbUsuarios()

{

List<ObjUsuario> lista = new List<ObjUsuario>();

conexion = Conexion.conexionBD();

conexion.Open();

comando = new NpgsqlCommand("SELECT nombre, usuario, fecha FROM public.usuarios", conexion);

NpgsqlDataReader dr = comando.ExecuteReader();

if (dr.HasRows)

{

while (dr.Read())

{

user = new ObjUsuario

{

nombre = dr.GetString(0),

usuario = dr.GetString(1),

fecha = dr.GetDateTime(2)

};

lista.Add(user);

}

}

conexion.Close();

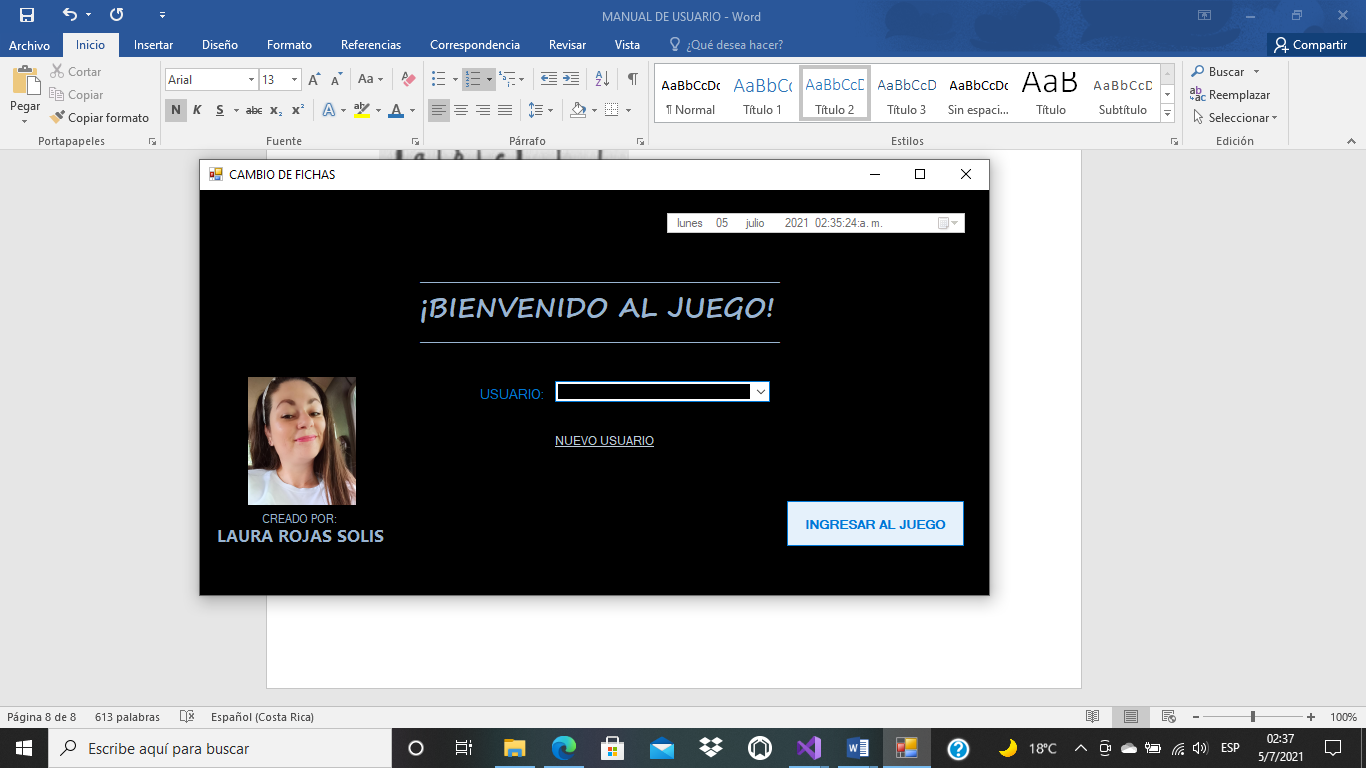
return lista;

}

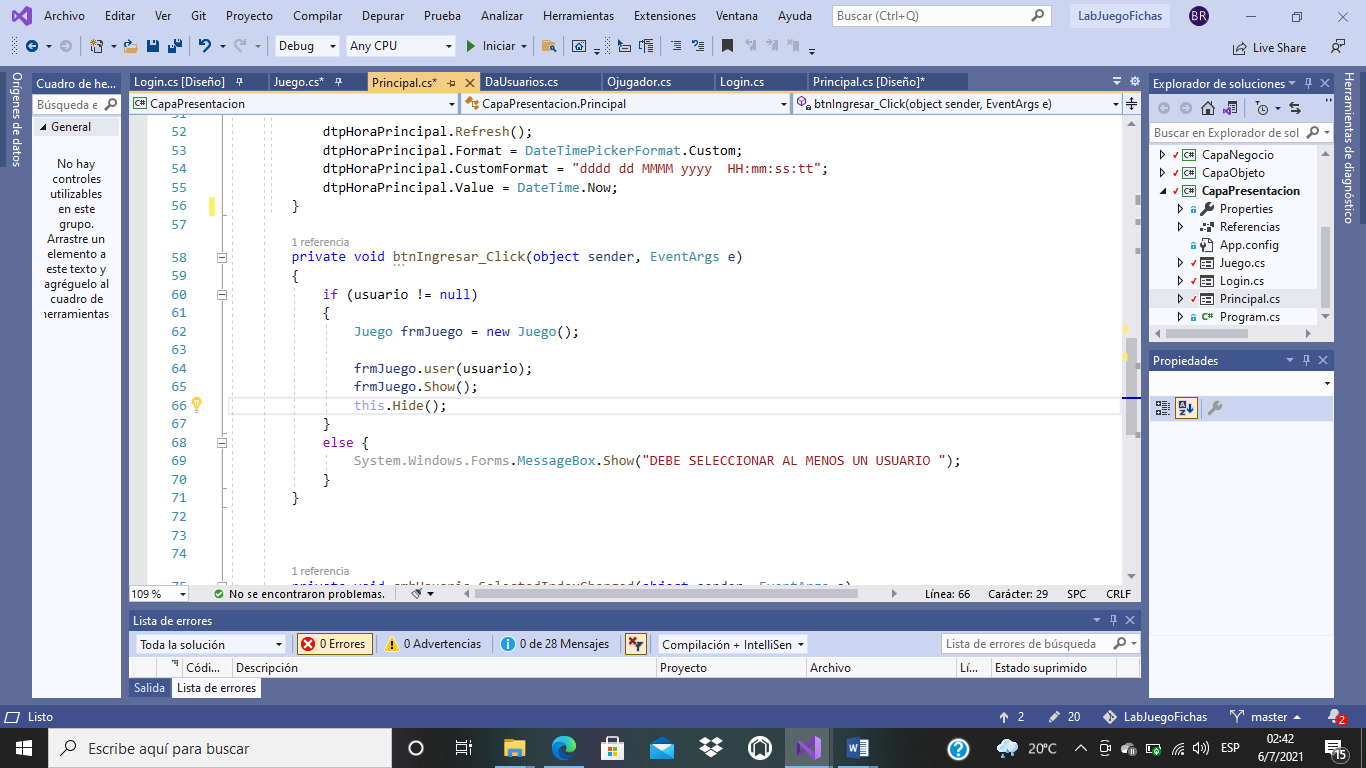
}

}

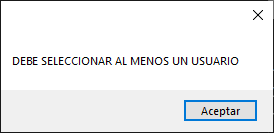
## **A button that allows access to the second system window.**



You create a form as game window form, and call to show this,, but remember that you needs select one player name, them we can write a condición to the gamer select your user´s name



If the user didn´t select a user, show a error window as this:



This is game form, to charge name user, we send it through a method like this:

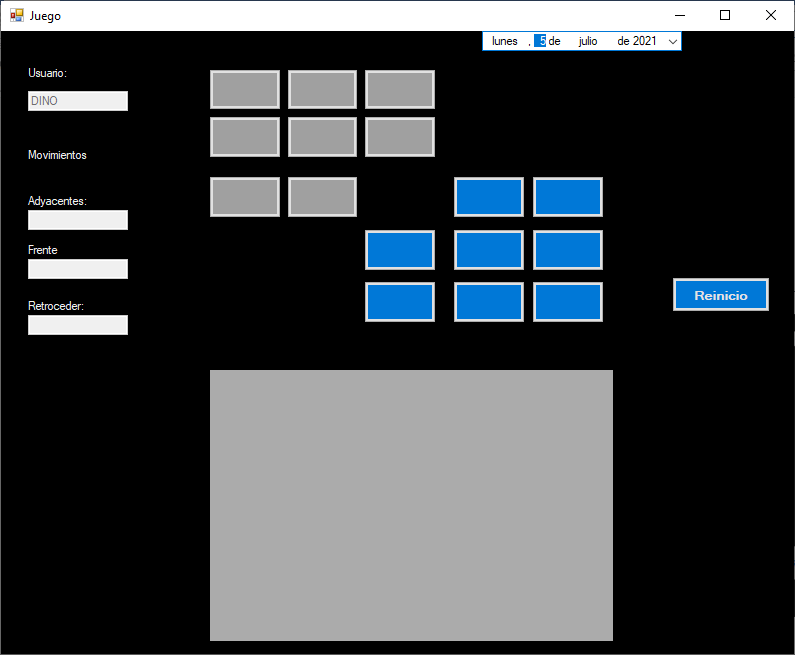
public void user(string user)

{

txtUsuario.Text = user;

usuario = txtUsuario.Text;

}



# **DATA BASE**

         Each time the game restarts or the user manages to find the solution, the following information must be saved in the database.

or User name.

o Date and time (when you registered to start the game).

o Number of adjacent moves.

or Number of hops.

o A field that is for the type of game whether you restarted it or found the solution.

         Some examples of the information that is saved:

or Alejandro | 12/02/2021 6:00 PM | 33 | 12 | Restart

or Alejandro | 12/02/2021 6:32 PM | 21 | 12 | Restart

or Alejandro | 14/02/2021 7:32 PM | 40 | 18 | Resolved

**Create the database on Postgrees this way:**

-- Database: juego

-- DROP DATABASE juego;

CREATE DATABASE juego

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'Spanish\_Costa Rica.1252'

LC\_CTYPE = 'Spanish\_Costa Rica.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

**Create one table to control game information on Postgrees this way:**

-- Table: public.control

-- DROP TABLE public.control;

CREATE TABLE public.control

(

usuario character varying COLLATE pg\_catalog."default",

fecha date,

adyacentes integer,

saltos integer,

tipo character varying COLLATE pg\_catalog."default"

)

TABLESPACE pg\_default;

ALTER TABLE public.control

OWNER to postgres;

**Create one table to user information on Postgrees this way:**

-- Table: public.usuarios

-- DROP TABLE public.usuarios;

CREATE TABLE public.usuarios

(

usuario character varying COLLATE pg\_catalog."default" NOT NULL,

nombre character varying COLLATE pg\_catalog."default",

fecha date,

CONSTRAINT usuarios\_pkey PRIMARY KEY (usuario)

)

TABLESPACE pg\_default;

ALTER TABLE public.usuarios

OWNER to postgres;

## **DASHBOARDS ON DATABASE**

GO

USE juego

go

--Data Select to any procedure

create proc DashboardDatos

@topganados int output,

@topadyacentes int output,

@contreinicio int output,

@contganado int output

as

Set topganados = (select count (tipo) as Ganados where tipo = 'Ganado' from control)

Set @topadyacentes = (select count(adyacentes) as MovimientoAdyacente from control)

Set @contreinicio =(select count(tipo) as Reiniciados where tipo = 'Reiniciado' from control)

Set contganado = (select count(tipo) as Ganados where tipo = 'Ganado' from control)

go

--Procedures

--Procedure to select 3 top user with Winner game amount

create proc TopGanados

as

Select Top 3 c.usuario+' '+c.tipo +' ' as TopGanador , COUNT(c.tipo) AS topGanador

from control

Group by topGanador,c.usuario

Order by count(3) desc

go

--Procedure to count who are the user with least amount of adjacent movements

create proc MenosAdyas

as

Select Top 3 c.usuario+' '+c.adyacentes+' ' as TopAdyac , COUNT(c.adyacentes) AS topAdyacentes

from control

Group by TopAdyac, c.usuario

Order by count(3) asc

go

* --Procedure to count how many times a user select reset control game.

create proc Resets

as

Select count (tipo ) AS Reiniciados WHERE (tipo = 'Reiniciado' )

from control

go

* --Procedure to compare how many times the people, global, reset the game.

create proc Goals

as

Select count (tipo ) AS Ganados WHERE (tipo = 'Ganado' )

from control

go