

Programación 3

Orientación a objetos con C#

5. Windows Forms

maurogullino@gmail.com

Nuevo proyecto

► Recientes

▲ Instalado

▲ Visual C#

Windows Universal
Escritorio clásico de Windows
.NET Core
.NET Standard
Prueba

► Otros lenguajes

► Otros tipos de proyectos

► En línea

¿No encuentra lo que busca?

[Abrir el instalador de Visual Studio](#)

.NET Framework 4.6.1

Ordenar por: Predeterminado



Buscar (Ctrl+E)



Aplicación vacía (Windows universal)

Visual C#



Aplicación de WPF (.NET Framework)

Visual C#



Aplicación de Windows Forms (.NET Framework)

Visual C#



Aplicación de consola (.NET Core)

Visual C#



Aplicación de consola (.NET Framework)

Visual C#



Biblioteca de clases (.NET Standard)

Visual C#



Biblioteca de clases (.NET Framework)

Visual C#



Proyecto compartido

Visual C#



Biblioteca de clases (portátil heredada)

Visual C#



Biblioteca de clases (Windows universal)

Visual C#

Tipo: Visual C#

Proyecto para crear una aplicación con una interfaz de usuario de Windows Forms

Nombre:

PruebaForms

Ubicación:

C:\Users\Mauro\source\repos

Nombre de la solución:

PruebaForms

Examinar...



Crear directorio para la solución

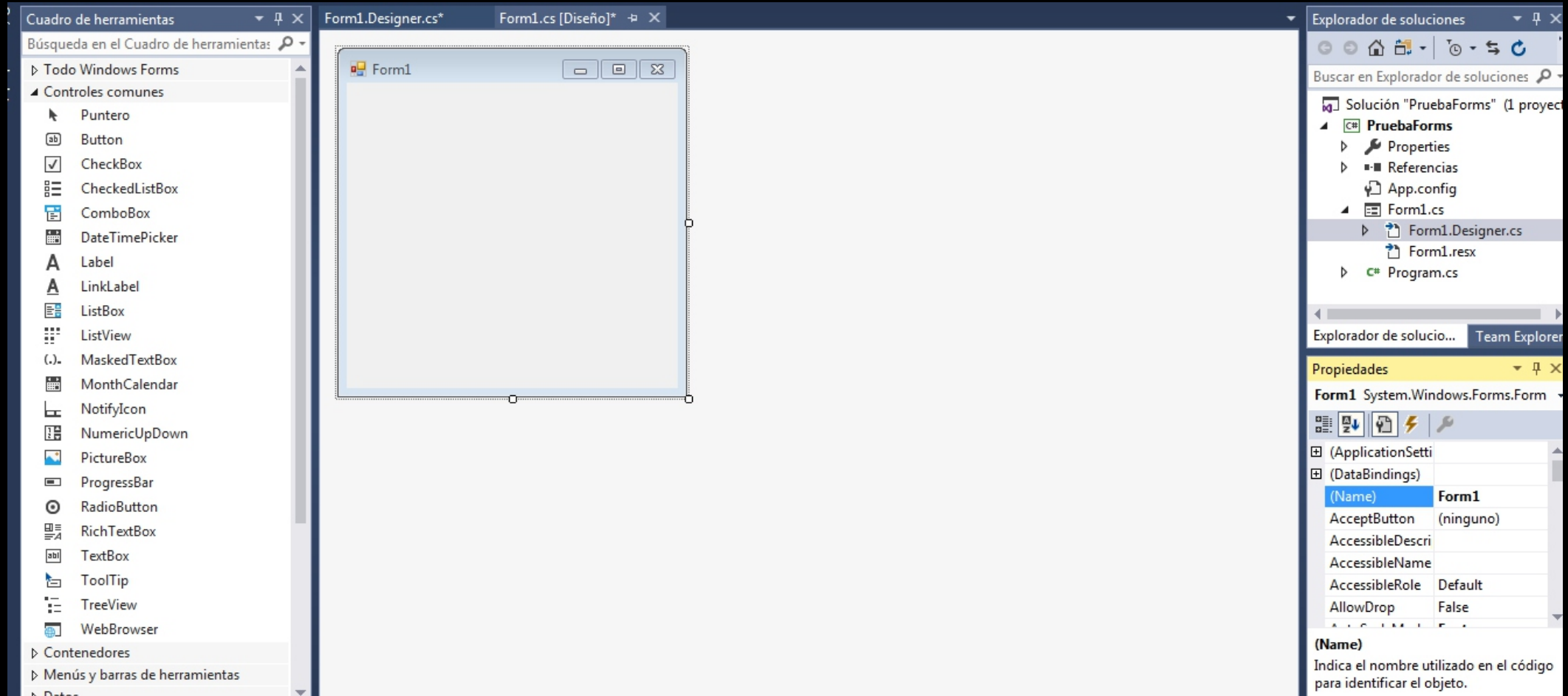


Agregar a control de código fuente

Aceptar

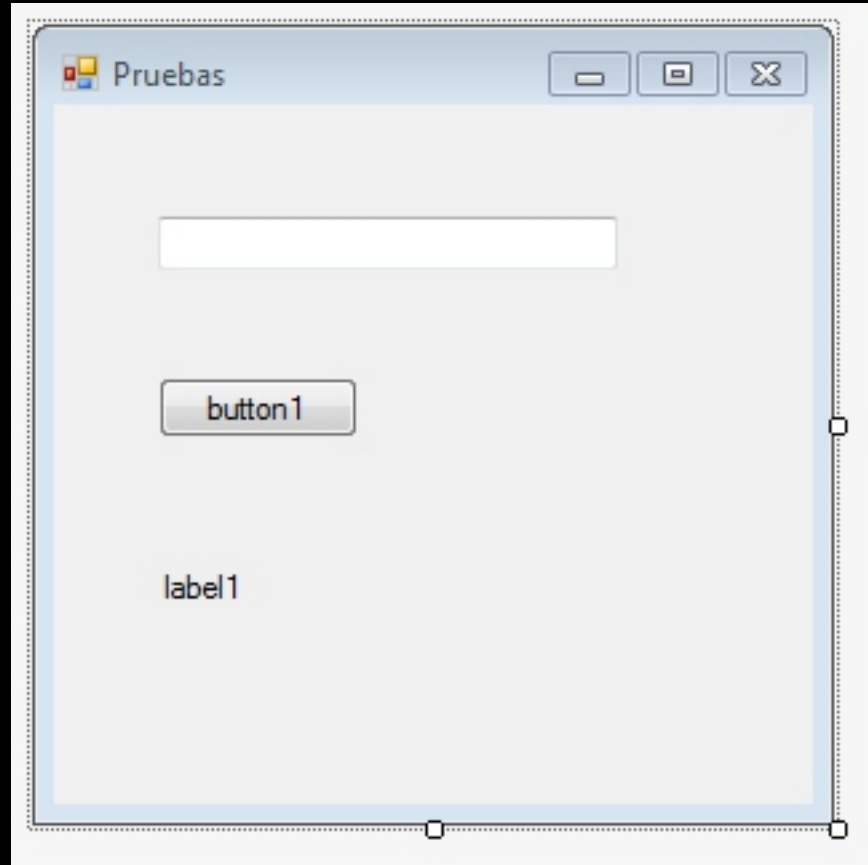
Cancelar

Vista de diseño



Cuadro de herramientas

Ventana Propiedades



FrmPrueba

Código generado por Designer

//frmPrueba.Designer.cs

```
namespace PruebaForms {  
    partial class FrmPrueba {  
  
        private System.ComponentModel.IContainer components = null;  
  
        private void InitializeComponent()  
        {  
            this.textBox1 = new System.Windows.Forms.TextBox();  
            this.button1 = new System.Windows.Forms.Button();  
            this.label1 = new System.Windows.Forms.Label();  
            // .....  
        }  
  
        private System.Windows.Forms.TextBox textBox1;  
        private System.Windows.Forms.Button button1;  
        private System.Windows.Forms.Label label1;
```

Código a modificar por nosotros

```
//frmPrueba.cs
```

```
using System.Windows.Forms;
```

```
namespace PruebaForms
```

```
{
```

```
    public partial class FrmPrueba : Form
```

```
    {
```

```
        public FrmPrueba()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
    }
```

```
}
```

El querido main

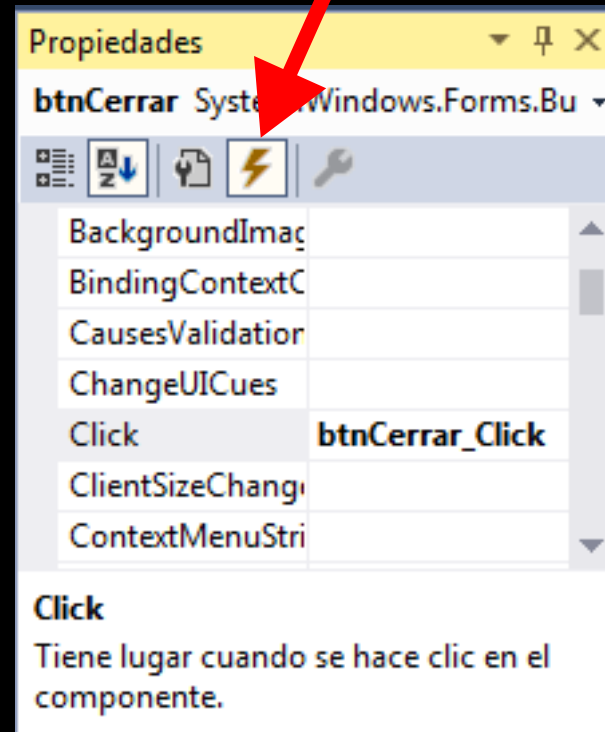
```
namespace PruebaForms
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FrmPrueba());
        }
    }
}
```

Agregamos un botón “cerrar”

```
public partial class FrmPrueba : Form
{
    public FrmPrueba()
    {
        InitializeComponent();
    }

    private void btnCerrar_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```


Eventos



MessageBox

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(textBox1.Text);
}
```

MessageBox más complejo

```
private void button1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("¿Escribió bien su nombre? " +
                        textBox1.Text, "Ejemplos",
                        MessageBoxButtons.YesNo,
                        MessageBoxIcon.Question)
        == DialogResult.Yes    )
    {
        MessageBox.Show("Lo felicito");
    }
    else {
        MessageBox.Show("Qué pena, corrija lo");
        textBox1.Text = "";
        textBox1.Focus();
    }
}
```

enumeraciones

```
enum MessageBoxButtons { YesNo, YesNoCancel, Ok, OkCancel };
```

```
enum DiaSemana { Lunes, Martes, Miercoles, Jueves,  
                Viernes, Sabado, Domingo };
```

```
//ejemplo  
public void AgregarEvento(DiaSemana dia, int hora, string tarea)  
{  
    //...  
}
```

```
agenda.AgregarEvento(DiaSemana.Martes, 20, "Pasear a Bobby");
```

Agregamos un TextBox para números

```
private void button1_Click(object sender, EventArgs e)
{
    int edad;
    if (!Int32.TryParse(txtEdad.Text, out edad)) {
        MessageBox.Show("Ingrese edad con números", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

        return;
    }

    if (edad < 18) {
        MessageBox.Show("Ud es menor de edad", "Alerta",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Botón oculto que se habilita con edad

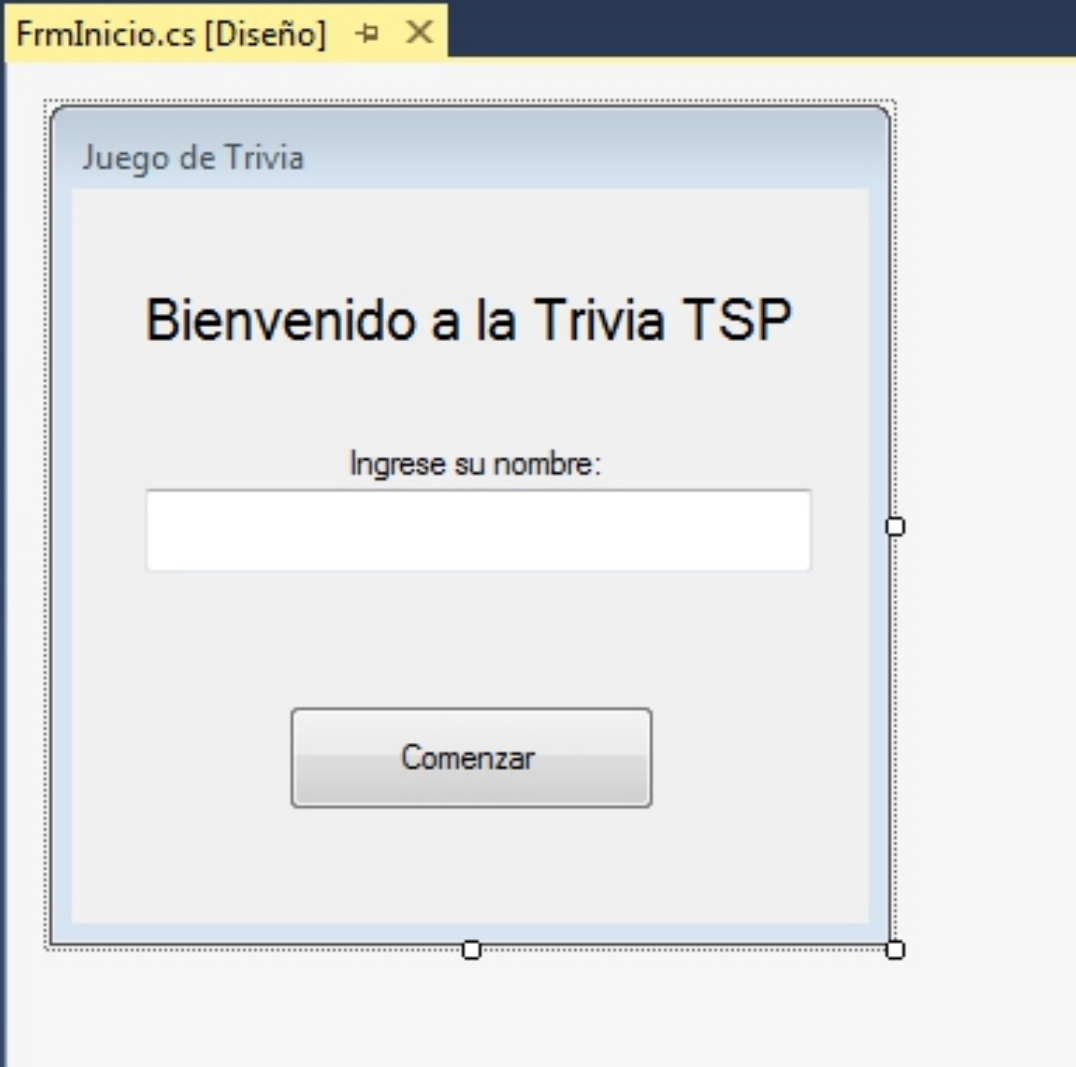
```
public partial class FrmPrueba : Form
{
    public FrmPrueba()
    {
        InitializeComponent();
        btnAcceder.Hide();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        // .....
        btnAcceder.Show();
    }
}
```

Ejemplo

Juego de Trivia

namespace JuegoTrivia



FrmInicio.cs [Diseño]

Juego de Trivia

Bienvenido a la Trivia TSP

Ingrese su nombre:

Comenzar

FrmInicio

txtNombre

btnComenzar

FrmJuego.cs*

FrmJuego.cs [Diseño]*

FrmInicio

FrmJuego

lblPregunta

☐ rbtOpcion1

☐ rbtOpcion2

☐ rbtOpcion3

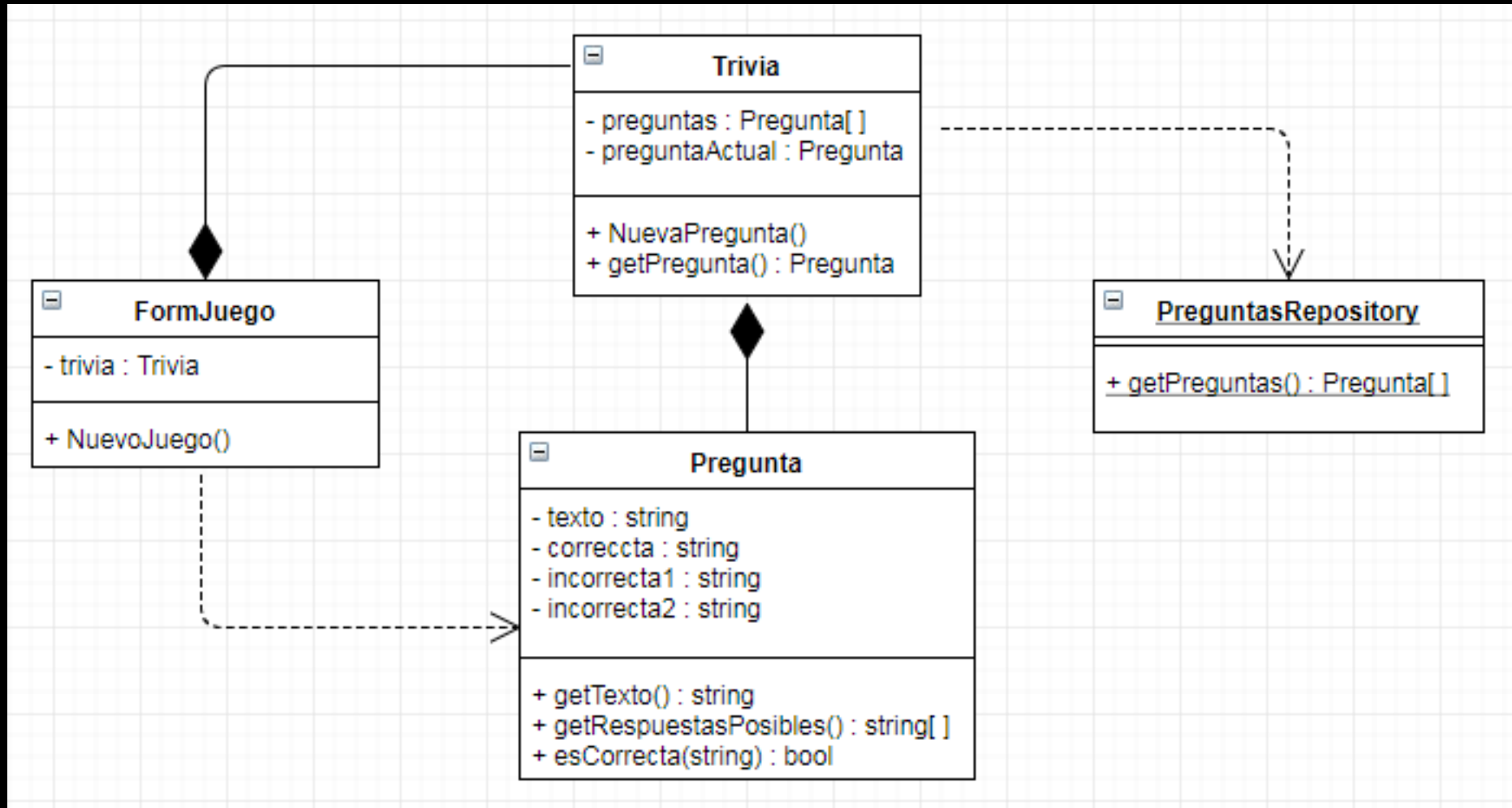
btnResponder

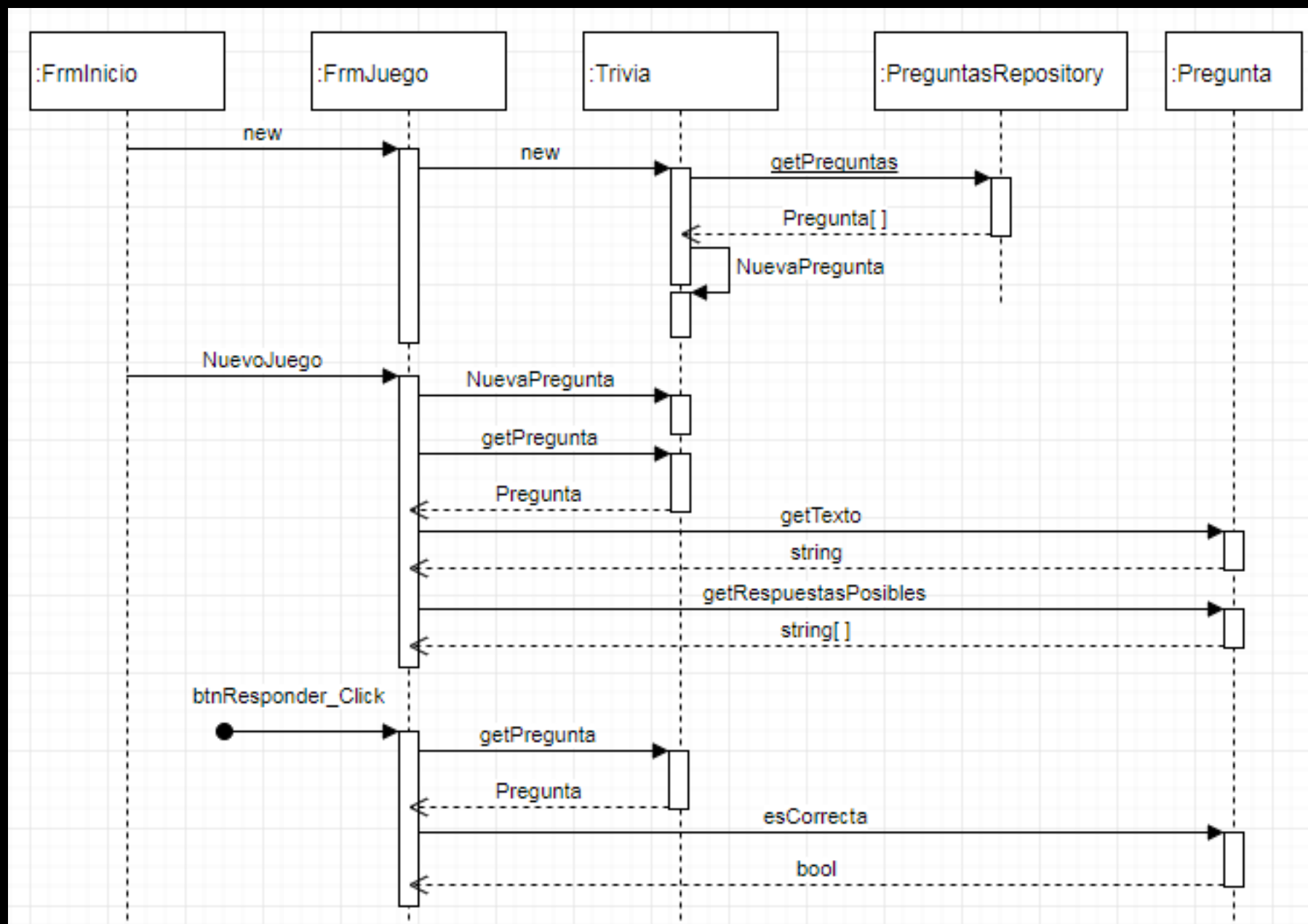
```
namespace JuegoTrivia {  
    public partial class FrmInicio : Form {  
        private FrmJuego frmJuego;  
  
        public FrmInicio()  
        {  
            InitializeComponent();  
            frmJuego = new FrmJuego(this);  
        }  
  
        private void btnComenzar_Click(object sender, EventArgs e)  
        {  
            if(txtNombre.Text.Length < 1) {  
                MessageBox.Show("Ingrese su nombre", "Error");  
                return;  
            }  
  
            frmJuego.Show();  
            this.Hide();  
        }  
    }  
}
```

```
namespace JuegoTrivia
{
    public partial class FrmJuego : Form
    {
        private Form frmInicio;

        public FrmJuego(Form inicio)
        {
            frmInicio = inicio;
            InitializeComponent();
        }
    }
}
```

Diseño





```
class Pregunta {
    private string texto;
    private string correcta;
    private string incorrecta1;
    private string incorrecta2;

    public Pregunta(string texto, string correcta, string incorrecta1,
                                                             string incorrecta2) {

        this.texto = texto;
        this.correcta = correcta;
        this.incorrecta1 = incorrecta1;
        this.incorrecta2 = incorrecta2;
    }

    public string getTexto() {    return texto;    }

    public string[] getRespuestasPosibles() {
        string[] respuestas = new string[] { correcta, incorrecta1, incorrecta2 };
        // .....
        return respuestas;
    }

    public bool esCorrecta(string respuesta) {
        if (respuesta == correcta) return true;
        else return false;
    }
}
```

```
public string[] getRespuestasPosibles() {  
    string[] respuestas = new string[] { correcta,  
                                         incorrecta1, incorrecta2 };  
  
    Random azar = new Random();  
  
    int n = respuestas.Length;  
    while(n > 1) //algoritmo Fisher-Yates (shuffle)  
    {  
        int indice = azar.Next(n--);  
        string temp = respuestas[n];  
        respuestas[n] = respuestas[indice];  
        respuestas[indice] = temp;  
    }  
  
    return respuestas;  
}
```

```
static class PreguntasRepository {  
    static public Pregunta[] getPreguntas()  
    {  
        Pregunta[] ret = new Pregunta[3];  
  
        ret[0] = new Pregunta("Cuánto mide el obelisco?",  
                                "68m", "45m", "83m");  
  
        ret[1] = new Pregunta("Cuál es la temperatura del sol?",  
                                "5778K", "1900K", "9541K");  
  
        ret[2] = new Pregunta("Qué es una neurona?",  
                                "Una célula", "Una mitocondria", "Un bioma");  
  
        return ret;  
    }  
}
```



```
class Trivia {
    private Pregunta[] preguntas;
    private Pregunta preguntaActual;

    public Trivia()
    {
        preguntas = PreguntasRepository.getPreguntas();
        NuevaPregunta();
    }

    public void NuevaPregunta()
    {
        Random azar = new Random();
        int cual = azar.Next(0, preguntas.Length);
        preguntaActual = preguntas[cual];
    }

    public Pregunta getPregunta()
    {
        return preguntaActual;
    }
}
```

[illegible]

//FrmJuego

```
public void NuevoJuego()
{
    trivia.NuevaPregunta();
    Pregunta p = trivia.getPregunta();
    lblPregunta.Text = p.getTexto();

    string[] respPosibles = p.getRespuestasPosibles();
    rbtOpcion1.Text = respPosibles[0];
    rbtOpcion2.Text = respPosibles[1];
    rbtOpcion3.Text = respPosibles[2];
}
```

//FrmJuego

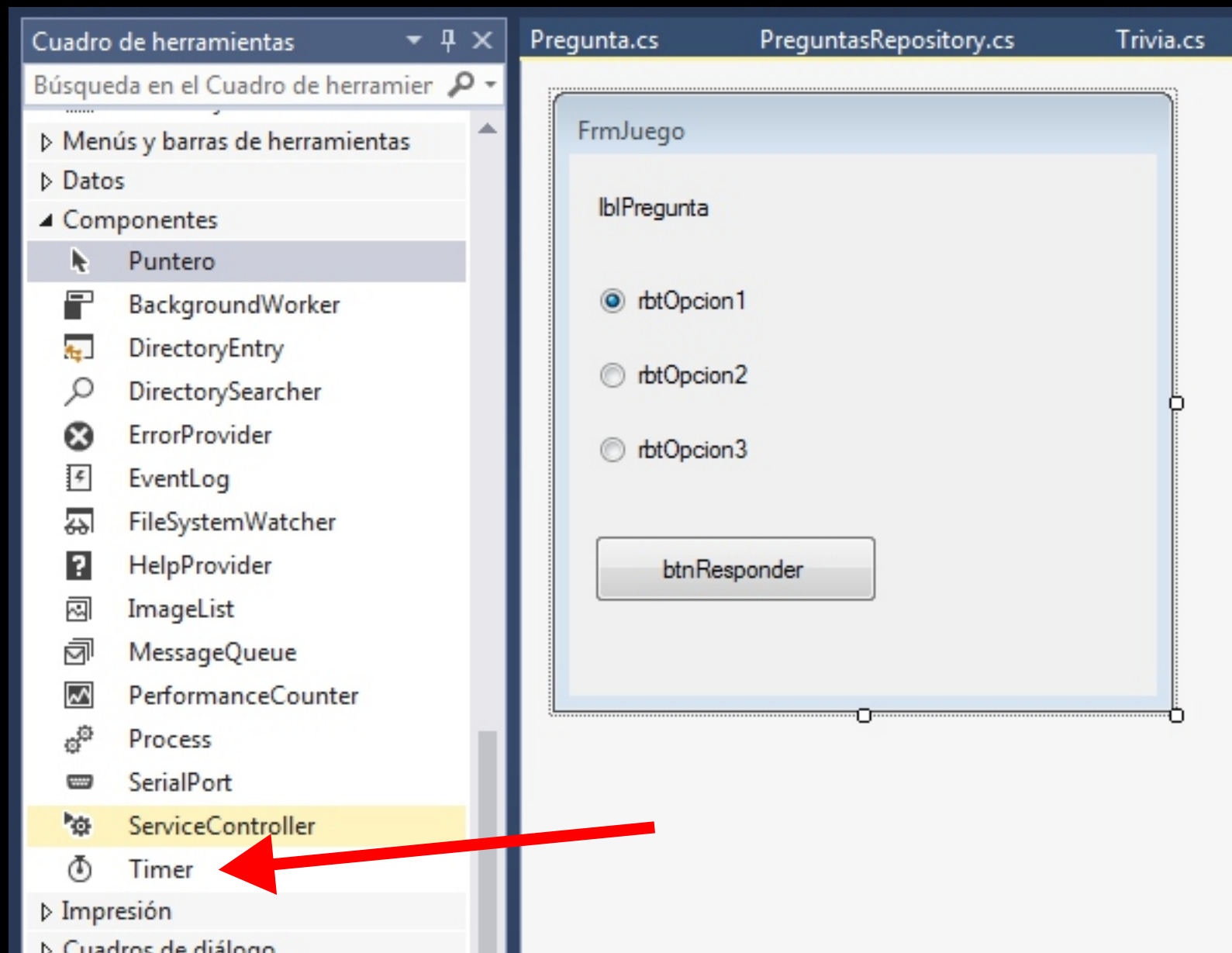
```
private void btnResponder_Click(object sender, EventArgs e)
{
    string respondida = "";
    if (rbtOpcion1.Checked) respondida = rbtOpcion1.Text;
    if (rbtOpcion2.Checked) respondida = rbtOpcion2.Text;
    if (rbtOpcion3.Checked) respondida = rbtOpcion3.Text;

    if(trivia.getPregunta().esCorrecta(respondida))
        MessageBox.Show("Correcto");
    else
        MessageBox.Show("MAL");

    NuevoJuego();
}
```

```
namespace JuegoTrivia {  
    public partial class FrmInicio : Form {  
        private FrmJuego frmJuego;  
  
        public FrmInicio()  
        {  
            InitializeComponent();  
            frmJuego = new FrmJuego(this);  
        }  
  
        private void btnComenzar_Click(object sender, EventArgs e)  
        {  
            if(txtNombre.Text.Length < 1) {  
                MessageBox.Show("Ingrese su nombre", "Error");  
                return;  
            }  
  
            frmJuego.NuevoJuego();  
            frmJuego.Show();  
            this.Hide();  
        }  
    }  
}
```

Timers



Visual Studio interface showing the design of a game application.

Form Design (FrmJuego):

- Form Title: FrmJuego
- Controls:
 - lblPregunta
 - Radio buttons: ☒ rbtOpcion1, ☐ rbtOpcion2, ☐ rbtOpcion3
 - Button: btnResponder
 - Label: 0 (indicated by a red arrow and the text **lblTiempo**)

Properties Window:

timer1 System.Windows.Forms.Timer

Property	Value
(Name)	timer1
Enabled	False
GenerateMemb	True
Interval	1000
Modifiers	Private
Tag	

Interval
Frecuencia de los eventos Elapsed en milisegundos.

Explorador de soluciones:

- Solución "JuegoTrivia" (1 proyecto)
- C# JuegoTrivia
 - Properties
 - Referencias
 - App.config
 - FrmInicio.cs
 - FrmInicio.Designer.cs
 - FrmInicio.resx

//FrmJuego

```
private void timer1_Tick(object sender, EventArgs e)
{
    lblTiempo.Text = (Int32.Parse(lblTiempo.Text)
                      + 1).ToString();
}
```

//FrmJuego

```
public void NuevoJuego()
{
    trivia.NuevaPregunta();
    Pregunta p = trivia.getPregunta();
    lblPregunta.Text = p.getTexto();

    rbtOpcion1.Text = p.getRespuestasPosibles()[0];
    rbtOpcion2.Text = p.getRespuestasPosibles()[1];
    rbtOpcion3.Text = p.getRespuestasPosibles()[2];

    lblTiempo.Text = "0";
    timer1.Enabled = true;
}
```

Ejercicio

- Agregar labels para indicar:
 - a) Puntos (cantidad de respuestas correctas)
 - b) Vidas (cantidad de errores, pierdo al llegar de 3 a 0)
- Agregar un tiempo límite para contestar
Luego de ese tiempo se cambia la pregunta y se contabiliza como error
- Al perder vuelvo al form inicial y puedo recomenzar