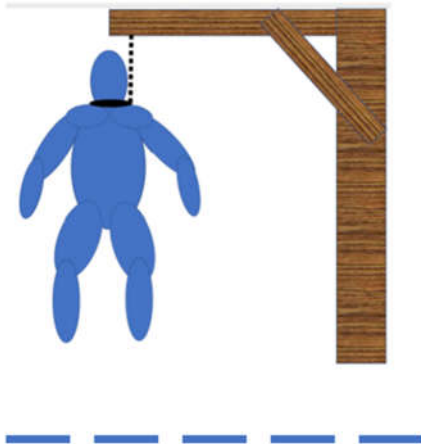


Hola, hoy vamos a hacer un juego muy popular: El ahorcado.

Primero vamos a ver un boceto de lo que se pretende hacer

Aquí la palabra que se va a adivinar

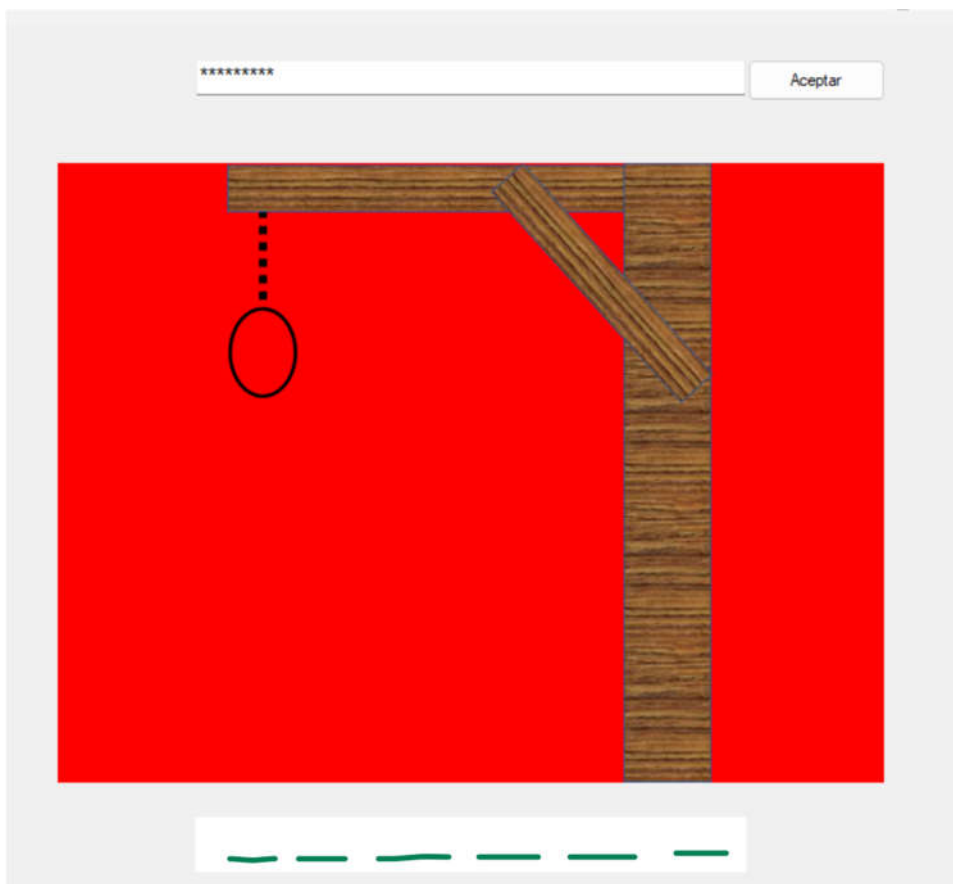


Bueno, primero los requerimientos.

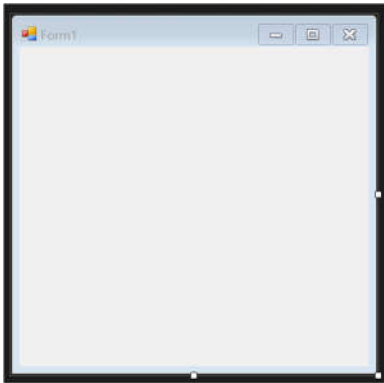
Debe tener un TextBox, para introducir la palabra, un panel que va a contener la imagen (podría ser un PictureBox) y un Label para la línea con guiones y que vaya cambiando a medida que se van adivinando las letras.

Yo puse un botón para que se iniciara la acción, sin embargo, una vez que termines y entiendas cómo funciona, podrás hacer tu propia implementación.

Aquí esta la interface grafica que les propongo.

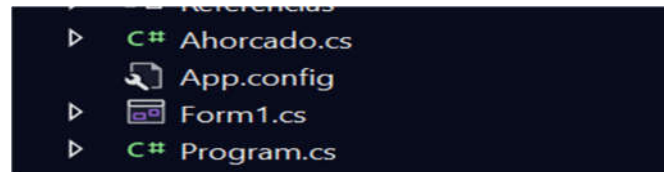


Bueno eso es lo que queremos, lo que tenemos es esto. Un form vacío



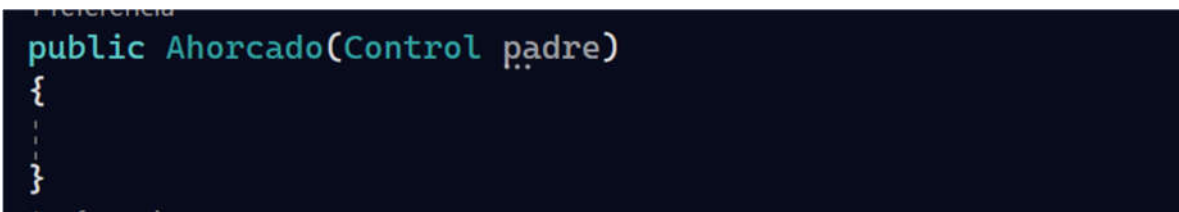
esto quiere decir que los controles y las interacciones de los objetos las vamos a crear de forma dinámica.

Para lo cual vamos a crear un proyecto que tenga un formulario llamado Form1 y una clase de nombre Ahorcado



Empezando

En la clase Ahorcado creamos un constructor **Ahorcado (Control padre)** con un parámetro de tipo Control que es la clase de la que heredan todos los controles con interface grafica.



Donde padre será el formulario donde se van a mostrar los objetos que vamos a crear. Además, vamos a crear los demás objetos de la interfaz como propiedades de la clase Ahorcado.



Bueno ahora si vamos a llenar el constructor de la clase Ahorcado, con la construcción de los objetos declarados, ojo que en el constructor que ya declaramos

```
1 referencia
public Ahorcado(Control padre)
{
    Padre = padre;

    Palabra = new TextBox
    {
        Name = "PalabraTextBox",
        Size = new System.Drawing.Size(400, 25),
        Location = new System.Drawing.Point(200, 25),
        Font = new System.Drawing.Font("Arial", 12),
        Text = "",
        PasswordChar = '*',
        Parent = Padre
    };
};
```

La propiedad **Padre** como pasa por parámetro simplemente se iguala, el TextBox se instancia

```
Aceptar = new Button
{
    Name="AceptarButton",
    Text="Aceptar",
    Location = new System.Drawing.Point(601, 25),
    Size = new System.Drawing.Size(100,30),
    Parent = Padre
};
```

Lo mismo el **Button**, la **Label** y el **Panel**

```
PalabraGuiones = new Label
{
    Name = "PalabraGuionesLabel",
    AutoSize = false,
    Size = new System.Drawing.Size(400, 40),
    Location = new System.Drawing.Point(200, 575),
    Font = new System.Drawing.Font("Arial", 16),
    BackColor = System.Drawing.Color.White,
    TextAlign = System.Drawing.ContentAlignment.MiddleCenter,
    Text = "",
    Parent = Padre
};
```

```

Contenedor = new Panel
{
    Name = "PalabraGuionesPanel",
    Size = new System.Drawing.Size(600, 450),
    Location = new System.Drawing.Point(100, 100),
    Font = new System.Drawing.Font("Arial", 12),
    BackColor = System.Drawing.Color.Red,
    BackgroundImage = Image.FromFile(@"img\cadalso.png"),
    BackgroundImageLayout= ImageLayout.Zoom,
    Parent = Padre
};

```

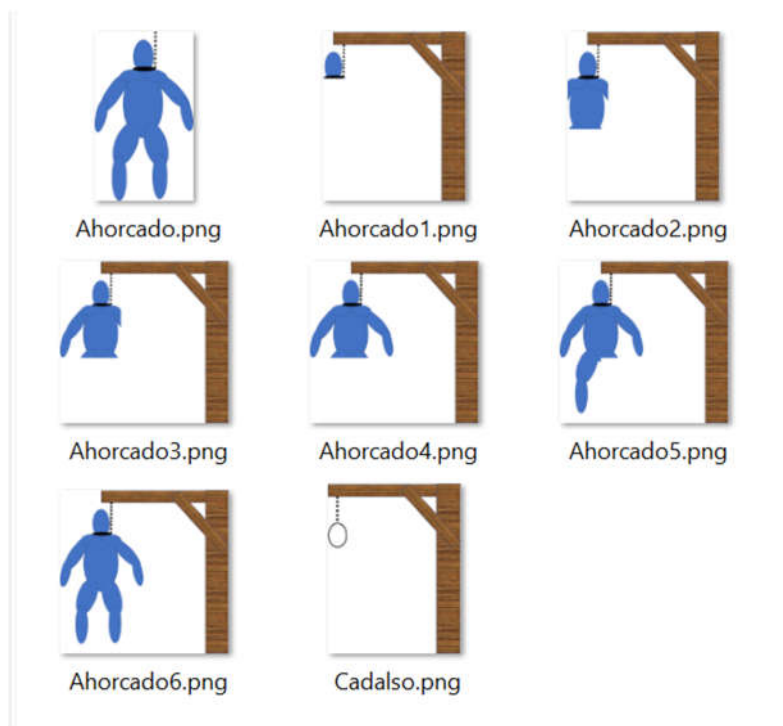
Como se puede ver el panel va a mostrar una imagen que viene de un archivo **"Cadalso.png"** que está en una carpeta llamada **img** y se muestra a escala, eso está en el código en las líneas

```

BackgroundImage = Image.FromFile(@"img\cadalso.png"),
BackgroundImageLayout= ImageLayout.Zoom,

```

Las imágenes que vamos a usar están en la carpeta raíz de la aplicación que es el **Debug** en este caso. Además, las imágenes que vamos a usar son las siguientes (las hice en PowerPoint)



Como vemos hay una imagen diferente para cada acción que se vaya a realizar, estas las hice yo, te invito a crear tus propias imágenes (realmente estas cumplen, pero son muy malitas).

Terminamos con el constructor de la Clase **Ahorcado**, agregando los eventos que se van a usar para la interacción del botón, para iniciar el juego y el evento en el formulario que captura las teclas que vamos presionando

```
Aceptar.Click += Aceptar_Click;  
Padre.Size=new System.Drawing.Size(800, 1000);  
  
padre.KeyDown += Padre_KeyDown;  
}
```

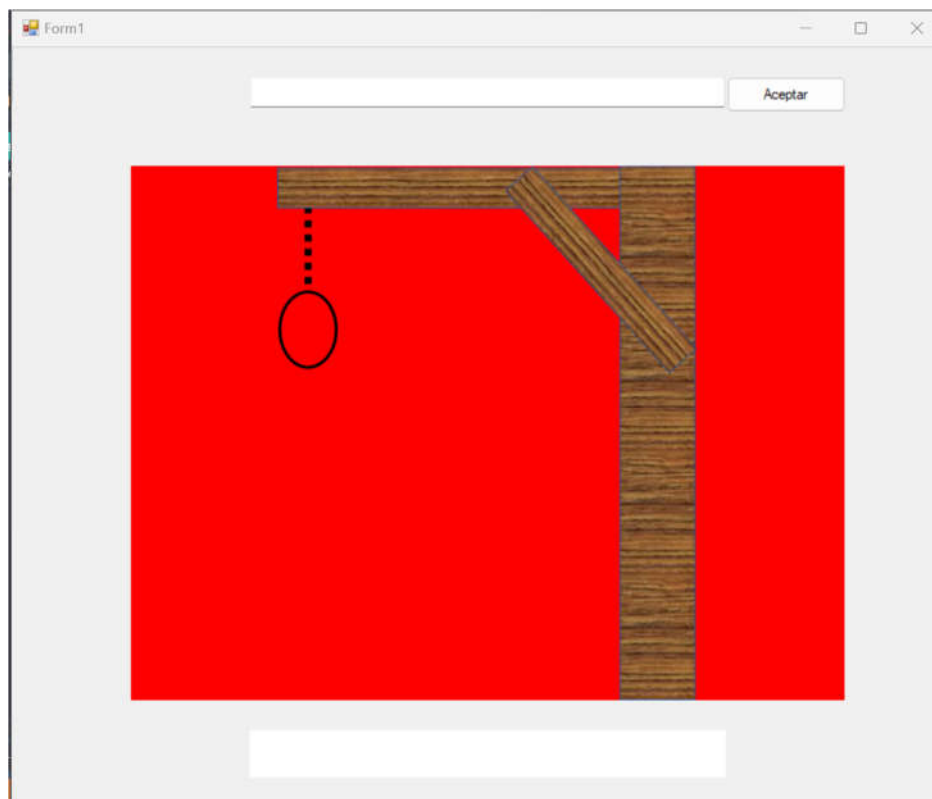
Al escribir `Aceptar.Click +=` da doble TAB y el IDE te completará el código de los eventos, sino te mostrará error hasta que crees los métodos

Te invito a consultar los **eventos del teclado en aplicaciones WindowsForm en .Net** y la propiedad `KeyPreview` de los formularios.

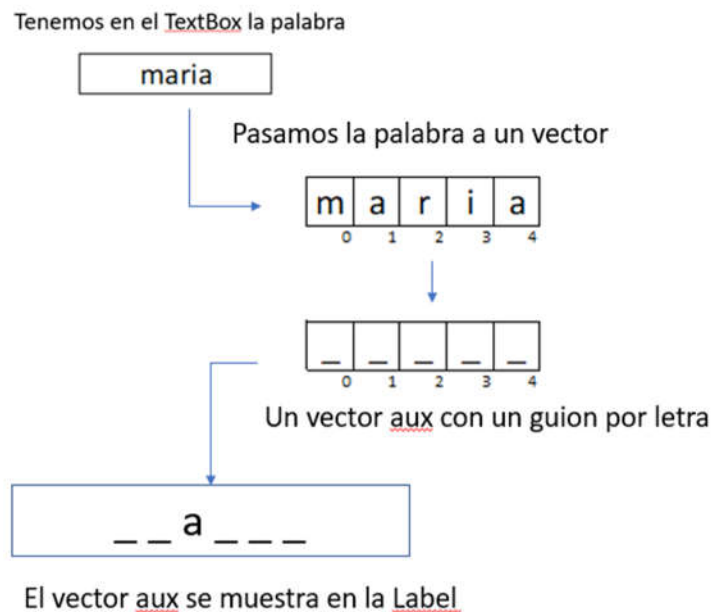
En el evento **Load** del Formulario **Form1** coloca el siguiente código y prueba.

```
2 referencias  
public void Form1_Load(object sender, EventArgs e)  
{  
    KeyPreview = true;  
    Ahorcado ahorcado = new Ahorcado(this);  
}
```

Debe aparecer algo como esto.



Ahora sí: el código duro, hasta ahora solo hemos creado la interfaz de usuario, ahora las interacciones. Primero la lógica de lo que vamos a hacer



La idea es tener un vector con las letras de la palabra y uno de igual tamaño pero lleno de guiones. Entonces, se busca la letra que corresponde a la tecla presionada en el vector de letras y si se encuentra, entonces se le asigna la letra a la en la posición correspondiente en el vector auxiliar, las veces que sea necesario.

Si la letra de la tecla presionada no se encuentra en el vector de letras entonces se le asigna al panel una de las imágenes que completan el ahorcado para esto se crea un contador que lleve la cuenta de las equivocaciones y de acuerdo con el contador se le asigna la imagen que corresponda.

Primero las variables necesarias. Deben ser variables de clase.

```
int index = 0;  
char[] chars;  
char[] charsGuiones;  
bool empezar=false;
```

Index => es el contador.

Chars => es el vector de letras.

charGuiones => vector auxiliar

empezar => para saber cuando empieza el juego, es falsa al comienzo, pero al dar clic al botón aceptar se vuelve verdadera.

Bueno ahora el evento click del botón aceptar

```
private void Aceptar_Click(object sender, EventArgs e)
{
    chars = Palabra.Text.ToCharArray();
    charsGuiones = new char[chars.Length];

    for (int i = 0; i < charsGuiones.GetLength(0); i++)
    {
        charsGuiones[i] = '_';
    }

    PrintGuiones();

    empezar = true;
    Palabra.Visible = false;
    Aceptar.Visible = false;
}
```

Luego el evento KeyDown

```
private void Padre_KeyDown(object sender, KeyEventArgs e)
{
    if (empezar)
    {
        if (chars.Contains(char.ToLower((char)e.KeyCode)))
        {
            for (int i = 0; i < chars.GetLength(0); i++)
            {
                char c = (char)e.KeyCode;
                if (char.ToLower(c) == chars[i])
                {
                    charsGuiones[i] = chars[i];
                }
            }

            PrintGuiones();
            Ganar();
        }
        else
        {
            Perder();
        }
    }
}
```


Ahora los métodos auxiliares

2 referencias

```
private void PrintGuiones()
{
    PalabraGuiones.Text = "";
    for (int i = 0; i < charsGuiones.GetLength(0); i++)
    {
        PalabraGuiones.Text += charsGuiones[i] + " ";
    }
}
```

1 referencia

```
private void Perder()
{
    if (index < 6)
    {
        Contenedor.BackgroundImage = Image.FromFile(@"img\ahorcado" + ++index + ".png");

        if (index == 6)
            MessageBox.Show("Perdiste pingo");
    }
}
```

1 referencia

```
private void Ganar()
{
    if (!charsGuiones.Contains('_'))
    {
        DialogResult result;
        result = MessageBox.Show("Ganaste", "Sistema", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (result == DialogResult.Yes)
        {
            Padre.Controls.Clear();
            ((Form1)Padre).Form1_Load(Padre, new EventArgs());
        }
    }
}
```

Y... listo ya tenemos el programa funcionando, espero que les haya gustado, lo hayan entendido y lo hayan hecho, preguntas al WhathApp