

JOGO DA FORCA

CLASE Form1

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;

namespace JogoForca_
{
    public partial class Form1 : Form
    {
        string palavraSecreta;
        char[] palavraOculto;
        List<char> letrasTentadas = new List<char>();
        int tentativasRestantes = 6;
        List<string> palavras = new List<string>
        {
            "ABACAXI", "AVIÃO", "BANANA", "CACHORRO", "CADERNO", "CARRO",
            "CELULAR", "COMPUTADOR", "ELEFANTE", "ESCOLA", "ESTRELA",
            "GIRAFÁ", "JANELA", "LIVRO", "MONTANHA", "PIANO",
            "PRAIA", "RELÓGIO", "SORVETE", "TARTARUGA"
        };
        bool jogoAtivo = false;

        public Form1()
        {
            InitializeComponent();

            private void btnNovoJogo_Click(object sender, EventArgs e)
            {
                Random rnd = new Random();
                int index = rnd.Next(palavras.Count);
                palavraSecreta = palavras[index];

                palavraOculto = new char[palavraSecreta.Length];
                for (int i = 0; i < palavraSecreta.Length; i++)
                {
                    palavraOculto[i] = '_';
                }

                letrasTentadas.Clear();
                tentativasRestantes = 6;
                lstLetrasTentadas.Items.Clear();
                lblPalavraOculto.Text = ExibirPalavraOculto();
                pnlPersonagem.Invalidate();
                jogoAtivo = true;
            }

            private void btnTentarLetra_Click(object sender, EventArgs e)
            {
                if (!jogoAtivo)
                {
                    MessageBox.Show("Clique em 'Novo Jogo' para começar.");
                    return;
                }

                string entrada = txtEntrada.Text.ToUpper();
                txtEntrada.Clear();

                if (entrada.Length != 1 || !char.IsLetter(entrada[0]))
```

```

    {
        MessageBox.Show("Digite apenas uma letra.");
        return;
    }

    char letra = entrada[0];

    if (letrasTentadas.Contains(letra))
    {
        MessageBox.Show("Letra já tentada.");
        return;
    }

    letrasTentadas.Add(letra);
    lstLetrasTentadas.Items.Add(letra);

    if (palavraSecreta.Contains(letra))
    {
        for (int i = 0; i < palavraSecreta.Length; i++)
        {
            if (palavraSecreta[i] == letra)
            {
                palavraOculto[i] = letra;
            }
        }
        lblPalavraOculto.Text = ExibirPalavraOculto();
    }
    else
    {
        tentativasRestantes--;
        pnlPersonagem.Invalidate();

        if (tentativasRestantes == 0)
        {
            MessageBox.Show("Você perdeu! A palavra era: " + palavraSecreta);
            jogoAtivo = false;
        }
    }

    if (!palavraOculto.Contains('_'))
    {
        MessageBox.Show("Parabéns! Você venceu!");
        jogoAtivo = false;
    }
}

private string ExibirPalavraOculto()
{
    string palavraComEspacos = "";
    foreach (char letra in palavraOculto)
    {
        palavraComEspacos += letra + " ";
    }
    return palavraComEspacos.Trim();
}

private void pnlPersonagem_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Pen caneta = new Pen(Color.Black, 2);

    g.DrawLine(caneta, 10, 150, 100, 150);
    g.DrawLine(caneta, 50, 150, 50, 20);
    g.DrawLine(caneta, 50, 20, 100, 20);
    g.DrawLine(caneta, 100, 20, 100, 40);

    if (tentativasRestantes <= 5) g.DrawEllipse(caneta, 85, 40, 30, 30);
    if (tentativasRestantes <= 4) g.DrawLine(caneta, 100, 70, 100, 110);
}

```

```
        if (tentativasRestantes <= 3) g.DrawLine(caneta, 100, 80, 80, 100);
        if (tentativasRestantes <= 2) g.DrawLine(caneta, 100, 80, 120, 100);
        if (tentativasRestantes <= 1) g.DrawLine(caneta, 100, 110, 80, 130);
        if (tentativasRestantes <= 0) g.DrawLine(caneta, 100, 110, 120, 130);
    }
}
```