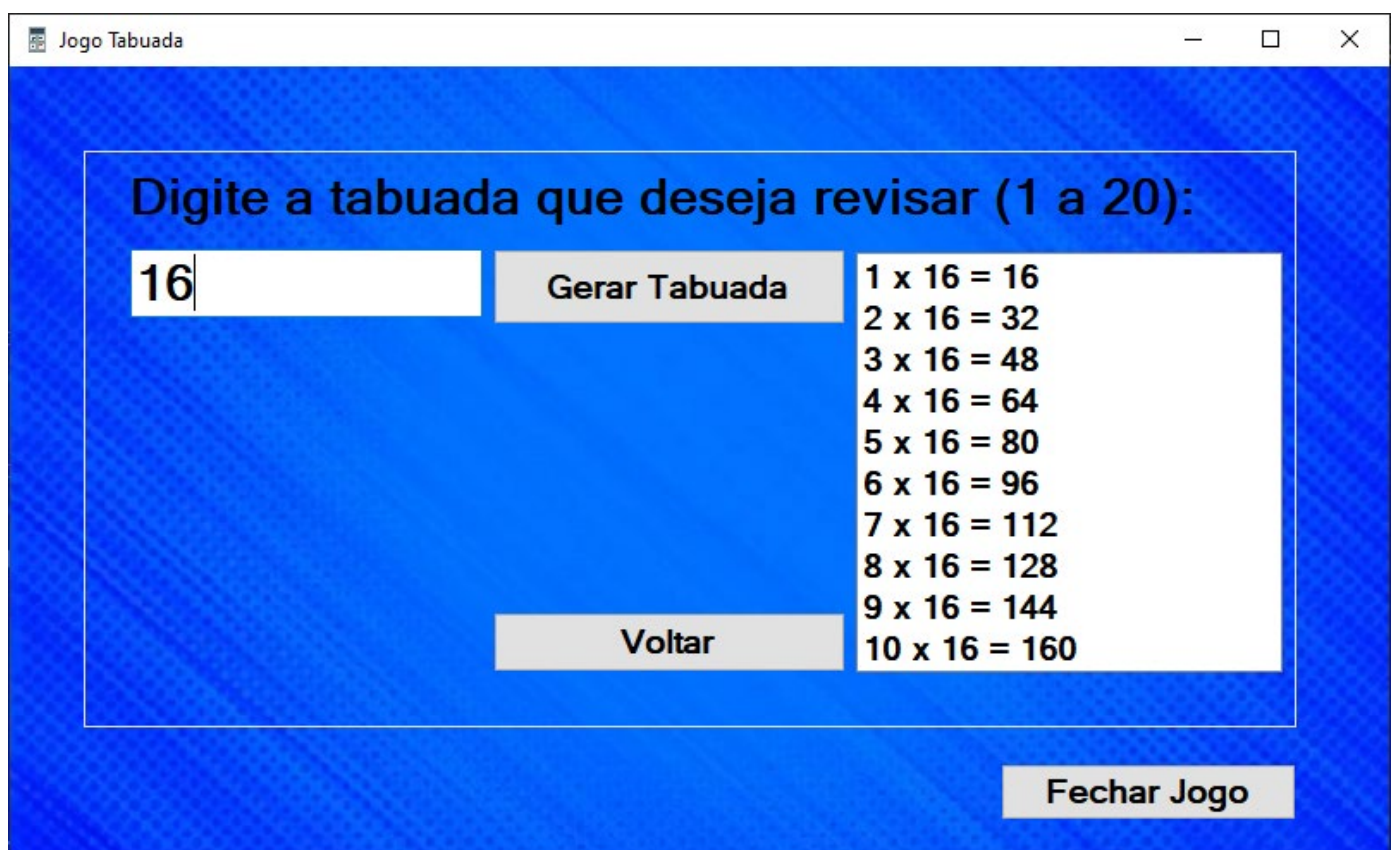
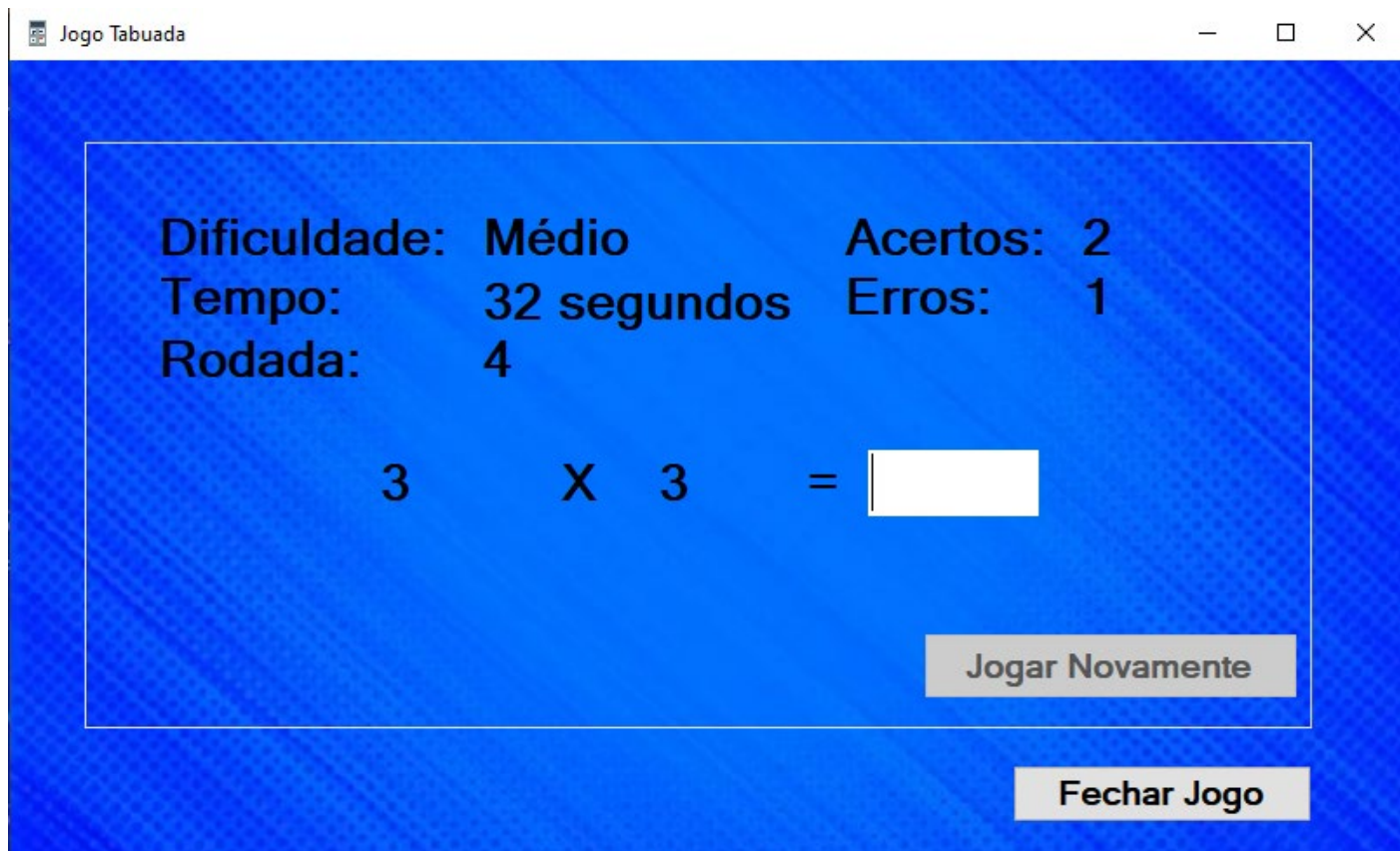
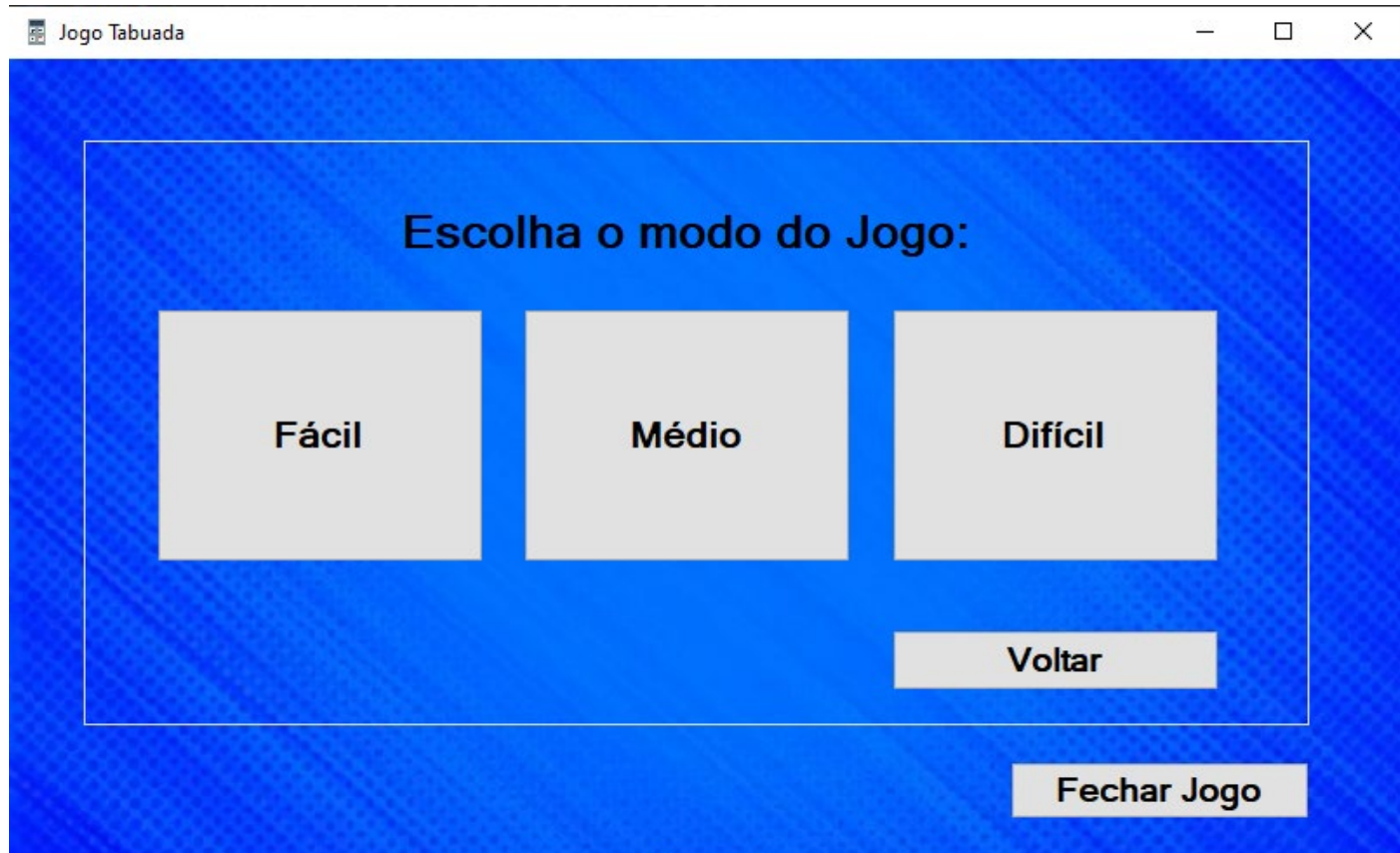


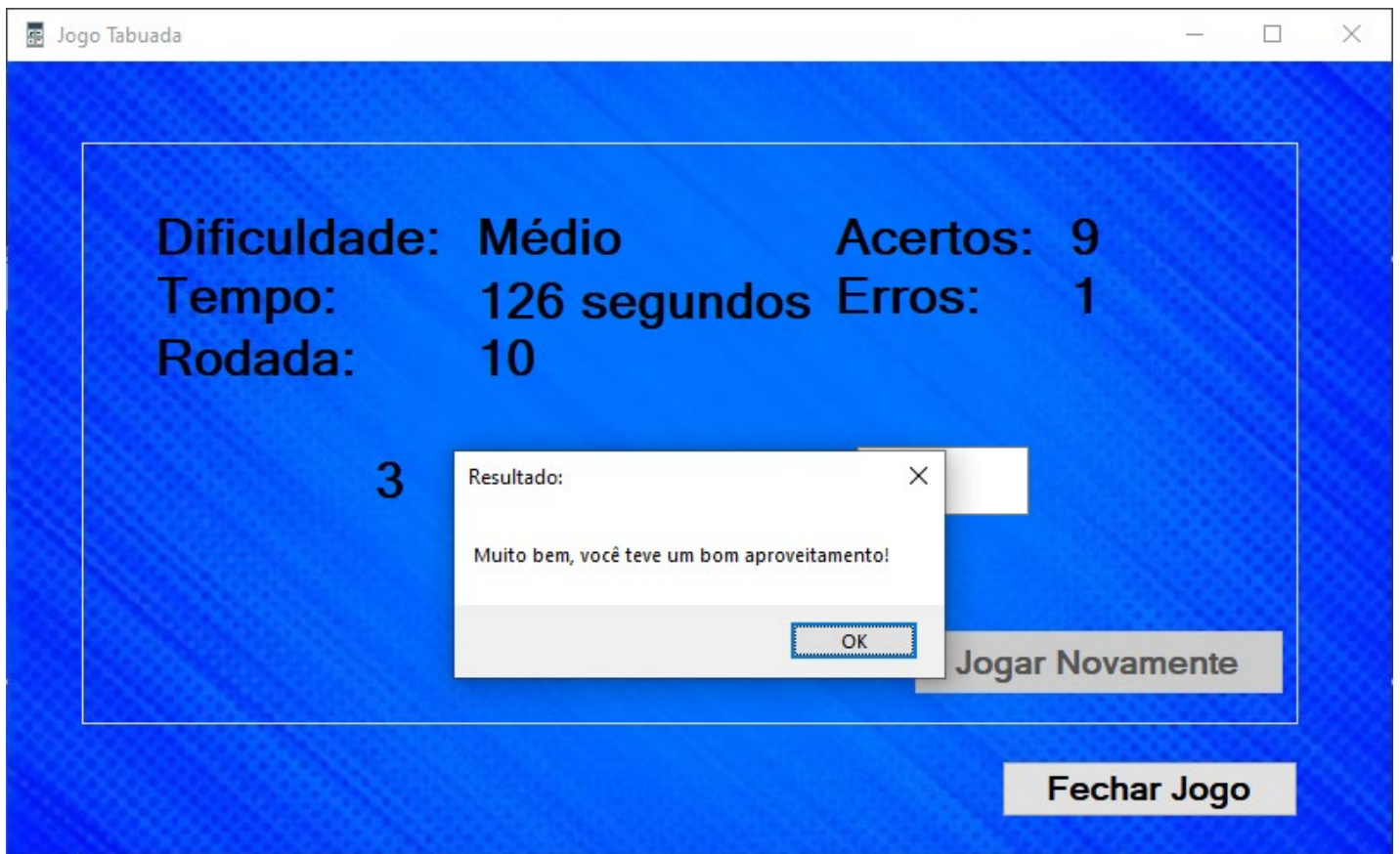
## Aplicativo Educacional: Jogo das Tabuadas

A função do Aplicativo é ser um jogo capaz de ajudar na memorização das tabuadas, divididas em modos: fácil (1 e 2), médio (3 a 10) e difícil (11 a 20).









Código:

```
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;

namespace AppEducativoTabuada
{
    public partial class Form1 : Form
    {
        int valor1, valor2, soma;
        int contador, acertos, erros, rodada = 0, rodadaFinal = 10;
        int ValorDigitado, i;
        private void btnRevisar_Click(object sender, EventArgs e)
        {
            gbInicio.Visible = false;
            gbRevisar.Visible = true;
            txtTabuada.Focus();
        }
    }
}
```

```
}
```

```
private void btnJogar_Click(object sender, EventArgs e)
{
    gbInicio.Visible = false;
    gbRevisar.Visible = false;
    gbModo.Visible = true;
}
```

```
private void btnFacil_Click(object sender, EventArgs e)
{
    gbModo.Visible = false;
    gbJogo.Visible = true;
    lblDificuldade.Text = "Fácil";
}
```

```
private void btnMedio_Click(object sender, EventArgs e)
{
    gbModo.Visible = false;
    gbJogo.Visible = true;
    lblDificuldade.Text = "Médio";
}
```

```
private void btnDificil_Click(object sender, EventArgs e)
{
    gbModo.Visible = false;
    gbJogo.Visible = true;
    lblDificuldade.Text = "Difícil";
}
```

```
e) private void btnJogarNovamente_Click(object sender, EventArgs
{
    gbModo.Visible = true;
    gbRevisar.Visible = true;
    gbInicio.Visible = true;
    btnComecar.Visible = true;
    lblDificuldade.Text = "";
    lblAcerto.Text = "";
    lblErro.Text = "";
    lblValor1.Text = "";
    lblValor2.Text = "";
    lblTempo.Text = "";
}
```

```

lblRodada.Text = "";
txtResultado.Text = "";
txtResultado.Enabled = false;
rodada = 0;
rodadaFinal = 10;
contador = 0;
acertos = 0;
erros = 0;
btnJogarNovamente.Enabled = false;
btnComecar.Focus();
lblAcerto.Text = "0";
lblErro.Text = "0";
}

```

```

private void txtTabuada_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8) {
        e.Handled = true;
    }
}

```

```

private void btnGerarTabuada_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    if (txtTabuada.Text == "")
    {
        MessageBox.Show("Digite um valor entre 1 a 20", "Erro:");
    }
    else
    {
        ValorDigitado = Convert.ToInt32(txtTabuada.Text);
        if (ValorDigitado >= 1 && ValorDigitado <= 20)
        {
            for (i = 1; i <= 10; i++)
            {
                listBox1.Items.Add(i + " x " + ValorDigitado + " = " + (i *
ValorDigitado));
            }
            txtTabuada.Text = "";
            txtTabuada.Focus();
        }
        else
        {
            txtTabuada.Text = "";

```

```

        txtTabuada.Focus();
        MessageBox.Show("Digite um valor entre 1 a 20", "Erro:");
    }
}

private void txtResultado_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != (char)8)
    {
        e.Handled = true;
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (timer1.Interval == 1000) {
        contador = contador + 1;
        lblTempo.Text = Convert.ToString(contador + " segundos");
    }
}

private void btnVoltar_Click(object sender, EventArgs e)
{
    gbRevisar.Visible = true;
    gbInicio.Visible = true;
    listBox1.Items.Clear();
    txtTabuada.Text = "";
}

private void btnVoltar2_Click(object sender, EventArgs e)
{
    gbRevisar.Visible = true;
    gbInicio.Visible = true;
}

private void txtResultado_KeyDown(object sender, KeyEventArgs
e)
{
    if (e.KeyCode == Keys.Enter) {
        Conferir();
        txtResultado.Focus();
    }
}

```

```
}
```

```
public Form1()
```

```
{
```

```
    InitializeComponent();
```

```
    btnComecar.Focus();
```

```
}
```

```
private void btnSair_Click(object sender, EventArgs e)
```

```
{
```

```
    Application.Exit();
```

```
}
```

```
private void NovaRodada()
```

```
{
```

```
    Random rdn = new Random();
```

```
    timer1.Enabled = true;
```

```
    while (rodada <= rodadaFinal)
```

```
    {
```

```
        if (lblDificuldade.Text == "Fácil")
```

```
        {
```

```
            valor1 = rdn.Next(1, 3);
```

```
            valor2 = rdn.Next(1, 10);
```

```
            lblValor1.Text = Convert.ToString(valor1);
```

```
            lblValor2.Text = Convert.ToString(valor2);
```

```
            txtResultado.Focus();
```

```
            rodada = rodada + 1;
```

```
        }
```

```
        else if (lblDificuldade.Text == "Médio")
```

```
        {
```

```
            valor1 = rdn.Next(3, 11);
```

```
            valor2 = rdn.Next(1, 10);
```

```
            lblValor1.Text = Convert.ToString(valor1);
```

```
            lblValor2.Text = Convert.ToString(valor2);
```

```
            txtResultado.Focus();
```

```
            rodada = rodada + 1;
```

```
        }
```

```
    else
```

```
    {
```

```
        valor1 = rdn.Next(11, 21);
```

```
        valor2 = rdn.Next(1, 10);
```

```
        lblValor1.Text = Convert.ToString(valor1);
```

```
        lblValor2.Text = Convert.ToString(valor2);
```

```
        txtResultado.Focus();
```

```

        rodada = rodada + 1;
    }
    break;
}
if(rodada > rodadaFinal)
{
    txtResultado.Enabled = false;
    timer1.Enabled = false;
    Final();
    lblValor1.Text = "";
    lblValor2.Text = "";
}
if (rodada < rodadaFinal)
{
    lblRodada.Text = Convert.ToString(rodada);
}
else {
    lblRodada.Text = Convert.ToString("10");
}
}

private void Final()
{
    if (acertos <= 3)
    {
        MessageBox.Show("Que pena, foi muito mal!",
"Resultado:");
    }
    else if(acertos > 3 && acertos <= 6)
    {
        MessageBox.Show("Você teve um aproveitamento
razoável!", "Resultado:");
    }
    else if (acertos > 6 && acertos <= 9)
    {
        MessageBox.Show("Muito bem, você teve um bom
aproveitamento!", "Resultado:");
    }
    else
    {
        MessageBox.Show("Parabéns, você não errou em nenhuma
rodada!", "Resultado:");
    }
}

```



```

        btnJogarNovamente.Enabled = true;

    }

    private void btnComecar_Click(object sender, EventArgs e)
    {
        btnComecar.Visible = false;
        txtResultado.Enabled = true;
        NovaRodada();
    }

    private void Conferir() {
        soma = valor1 * valor2;
        if (txtResultado.Text == Convert.ToString(soma))
        {
            acertos++;
            lblAcerto.Text = Convert.ToString(acertos);

        }
        else {
            erros++;
            lblErro.Text = Convert.ToString(erros);
        }
        txtResultado.Text = "";
        NovaRodada();
    }

}

```