

```

using System;
using System.IO;
using System.Windows.Forms;

namespace WindowsFormsApp
{
    public partial class FormVisualizarQuestao : Form
    {
        private string arquivoQuestoes = "questoes.txt"; // Nome do arquivo onde as questões
        estão armazenadas
        private int tempoPorMateria = 20; // Tempo em minutos por matéria
        private Timer timer;
        private int tempoRestante;

        public FormVisualizarQuestao()
        {
            InitializeComponent();
        }

        private void FormVisualizarQuestao_Load(object sender, EventArgs e)
        {
            // Preencher a ComboBox com as opções de matéria
            cmbMateria.Items.Add("Matemática");
            cmbMateria.Items.Add("Ciências");
            cmbMateria.Items.Add("História");
            cmbMateria.Items.Add("Geografia");
            // Adicione mais matérias, se necessário
        }

        private void cmbMateria_SelectedIndexChanged(object sender, EventArgs e)
        {
            // Carregar a primeira questão da matéria selecionada
            CarregarQuestao();
            // Iniciar a contagem de tempo
            IniciarContagemTempo();
        }

        private void btnProxima_Click(object sender, EventArgs e)
        {
            // Salvar a resposta da questão atual antes de passar para a próxima questão
            SalvarResposta();
            // Carregar a próxima questão
            CarregarQuestao();
        }

        private void CarregarQuestao()
        {
            string materia = cmbMateria.SelectedItem?.ToString();

```

```

if (string.IsNullOrEmpty(materia))
    return;

try
{
    using (StreamReader sr = new StreamReader(arquivoQuestoes))
    {
        string line;
        bool encontrouQuestao = false;
        while ((line = sr.ReadLine()) != null)
        {
            if (line == materia)
            {
                // Encontrou uma questão da matéria selecionada
                encontrouQuestao = true;
                txtQuestao.Text = sr.ReadLine();
                rbOpcao1.Text = sr.ReadLine();
                rbOpcao2.Text = sr.ReadLine();
                rbOpcao3.Text = sr.ReadLine();
                break;
            }
        }

        if (!encontrouQuestao)
        {
            MessageBox.Show("Não há mais questões disponíveis para a matéria selecionada.", "Fim das questões", MessageBoxButtons.OK, MessageBoxIcon.Information);
            LimparCampos();
            PararContagemTempo();
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show("Ocorreu um erro ao carregar a questão: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void SalvarResposta()
{
    // Implemente a lógica para salvar a resposta da questão atual
    // Você pode armazenar as respostas em uma estrutura de dados ou em um arquivo,
    // por exemplo
}

private void LimparCampos()
{

```

```

        txtQuestao.Text = "";
        rbOpcao1.Text = "";
        rbOpcao2.Text = "";
        rbOpcao3.Text = "";
        rbOpcao1.Checked = false;
        rbOpcao2.Checked = false;
        rbOpcao3.Checked = false;
    }

    private void IniciarContagemTempo()
    {
        tempoRestante = tempoPorMateria;
        timer = new Timer();
        timer.Interval = 1000; // 1 segundo
        timer.Tick += Timer_Tick;
        timer.Start();
    }

    private void PararContagemTempo()
    {
        timer.Stop();
        timer.Dispose();
    }

    private void Timer_Tick(object sender, EventArgs e)
    {
        tempoRestante--;
        if (tempoRestante == 0)
        {
            MessageBox.Show("O tempo para a matéria selecionada esgotou.", "Tempo esgotado", MessageBoxButtons.OK, MessageBoxIcon.Information);
            LimparCampos();
            PararContagemTempo();
        }
    }
    // Restante do código...

    private void btnResponder_Click(object sender, EventArgs e)
    {
        VerificarResposta();
    }

    private void VerificarResposta()
    {
        string materia = cmbMateria.SelectedItem?.ToString();
        if (string.IsNullOrEmpty(materia))
            return;
    }

```

```

try
{
    using (StreamReader sr = new StreamReader(arquivoQuestoes))
    {
        string line;
        bool encontrouQuestao = false;
        while ((line = sr.ReadLine()) != null)
        {
            if (line == materia)
            {
                // Encontrou uma questão da matéria selecionada
                encontrouQuestao = true;
                string pergunta = sr.ReadLine();
                string respostaCorreta = sr.ReadLine();
                txtQuestao.Text = pergunta;
                rbOpcao1.Text = sr.ReadLine();
                rbOpcao2.Text = sr.ReadLine();
                rbOpcao3.Text = sr.ReadLine();

                // Verificar a resposta selecionada pelo usuário
                if (rbOpcao1.Checked && rbOpcao1.Text == respostaCorreta)
                {
                    lblResposta.Text = "Resposta correta!";
                }
                else if (rbOpcao2.Checked && rbOpcao2.Text == respostaCorreta)
                {
                    lblResposta.Text = "Resposta correta!";
                }
                else if (rbOpcao3.Checked && rbOpcao3.Text == respostaCorreta)
                {
                    lblResposta.Text = "Resposta correta!";
                }
                else
                {
                    lblResposta.Text = "Resposta incorreta!";
                }

                break;
            }
        }

        if (!encontrouQuestao)
        {
            MessageBox.Show("Não há mais questões disponíveis para a matéria selecionada.", "Fim das questões", MessageBoxButtons.OK, MessageBoxIcon.Information);
            LimparCampos();
            PararContagemTempo();
        }
    }
}

```

```
    }  
  }  
  catch (Exception ex)  
  {  
    MessageBox.Show("Ocorreu um erro ao carregar a questão: " + ex.Message,  
"Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);  
  }  
}  
}  
}  
}
```