

## Estrutura Switch/Case

Aprender a criar estruturas de condição é fundamental para programar a lógica de um sistema. Essas estruturas permitem a execução de diferentes linhas de código (diferentes fluxos de execução) a partir da avaliação de uma condição. Na linguagem **C#** temos dois recursos com esse propósito: `if/else` e `switch/case`.

### Switch/Case

Switch/case é uma estrutura de condição que define o código a ser executado com base em uma comparação de valores.

Para que isso fique mais claro, vejamos a sintaxe do `switch/case`:

```
switch (variável ou valor)
{
case valor1:
    // código 1
break;
case valor2:
    // código 2
break;
}
```

Na linha 1, em `switch (variável ou valor)`, definimos a variável ou valor que desejamos comparar. Na linha 3, informamos que se o valor declarado neste **case** for igual ao contido no **switch**, `código 1` será executado. O mesmo comportamento se aplica ao segundo `case`. Ademais, caso o valor contido no `switch` não seja atendido em uma das condições, nenhum bloco será executado.

E o comando `break`? O comando **break** é utilizado para especificar a última linha de código a ser executada dentro da condição. Se não declarado, os códigos implementados dentro dos **cases** subsequentes serão executados.

## Default

---

O operador `default` é utilizado quando precisamos definir um fluxo alternativo para as situações em que o valor contido no `switch` não seja atendido por nenhum dos **cases** especificados.

```
switch (variável ou valor)
{
case valor1:
    // código 1
break;
case valor2:
    // código 2
break;
default:
    // código 3
    break;
}
```

Caso o valor do `switch` não seja igual a um dos valores contidos nos **cases**, o sistema irá executar o código implementado no bloco `default`; neste exemplo, o código 3.

Projeto Exemplo:

Crie um projeto WindowsForm e monte o design a seguir:

The screenshot shows a Windows Form titled "Sistema de Ajuste Salarial". The form contains the following elements:

- Five text boxes for input:
  - Nome: [ ]
  - Código da Profissão: [ ]
  - Salário Bruto: [ ]
  - Profissão: [ ]
  - Salário Atualizado: [ ]
- A list box on the right side containing three items:
  - 1: Analista de Sistemas
  - 2: Programador
  - 3: Adm de Banco de Dados
- Three buttons at the bottom:
  - Calcular
  - Limpar
  - Fechar

**Names:**

TextBox txtNome;

TextBox txtcodigo;

TextBox txtsalario;

TextBox txtprofissao;

txtreaj;

Button btnCalcular

Button btnlimpar

Button btnFechar

Altere a propriedade ReadOnly (somente leitura) do txtprofissao e do txtreaj para TRUE, essa propriedade tem a função de não permitir a digitação em campos que serão utilizados para mostrar resultados.

Selecione o [textBox](#) txtcodigo, na janela de propriedades selecione os eventos (raio) e dê duplo clique em KeyPress

Digite as linhas em negrito:

```
private void txtcodigo_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar >= '0' && e.KeyChar <= '9' || e.KeyChar == 8)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
```

O exemplo anterior determina se o usuário pressionou uma tecla numérica ou a tecla backspace, em caso afirmativo, cancela o evento [KeyPress](#) usando a propriedade [Handled](#) (manipulado), isso significa que o textbox receberá a tecla pressionada.

Se você definir Handled como true em um [TextBox](#) , esse controle não passará o pressionamento de tecla, o que acontece dentro do else, quando for pressionada uma tecla não numérica.

Vamos programar os três buttons criados no form, dê dois cliques e programe o evento Click de cada botão.

```
private void btnCalcular_Click(object sender, EventArgs e)
{
    double salario, reajsal;
    int codigo;

    if (txtcodigo.Text == "" || txtsalario.Text=="")
    {
        MessageBox.Show("Digite");
    }
    else {
        salario = double.Parse(txtsalario.Text);
        codigo = Int32.Parse(txtcodigo.Text);

        switch (codigo)
        {
            case 1:
            {
                reajsal = salario * 1.10;
                txtreaj.Text = reajsal.ToString("C2");
                txtprofissao.Text = "Analista de Sistemas";
                break;
            }

            case 2:
            {
                reajsal = salario * 1.05;
                txtreaj.Text = reajsal.ToString("C2");
            }
        }
    }
}
```

```
        txtprofissao.Text = "Programador";
        break;

    }

    case 3:
    {
        reajsal = salario * 1.15;
        txtreaj.Text = reajsal.ToString("C2");
        txtprofissao.Text = "Adm de Banco de Dados";
        break;

    }

    default:
    {

        MessageBox.Show("Opção Inválida");
        break;

    }

}

private void btnlimpar_Click(object sender, EventArgs e)
{
    txtreaj.Clear();
    txtsalario.Clear();
    txtprofissao.Clear();
    txtcodigo.Clear();
    txtcodigo.Focus();
}
```

```
private void btnFechar_Click(object sender, EventArgs e)
{

    MessageBox.Show(txtnome.Text + "\nAté breve");
    Close();
}
```

### Controle C# PictureBox

O controle PictureBox do Windows Forms é usado para exibir imagens nos formatos bitmap, GIF, ícone ou JPEG.

Você pode definir a propriedade Image para a imagem que deseja exibir, seja em tempo de design ou em tempo de execução. Você pode alterar programaticamente a imagem exibida em uma caixa de imagem, o que é particularmente útil quando você usa um único formulário para exibir diferentes informações.

```
pictureBox1.Image = Image.FromFile("c:\\Users\\crisv\\Desktop\\imagem1.jpg");
```

A propriedade SizeMode, que é definida como valores na enumeração PictureBoxSizeMode, controla o recorte e o posicionamento da imagem na área de exibição.

```
pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
```

Há cinco PictureBoxSizeMode diferentes disponíveis para o controle PictureBox.

- AutoSize - Dimensiona a caixa de imagem para a imagem.
- CenterImage - Centraliza a imagem na caixa de imagem.
- Normal - Coloca o canto superior esquerdo da imagem no canto superior esquerdo da caixa de imagem
- StretchImage - Permite esticar a imagem no código

## Limpar o pictureBox

Para limpar um gráfico

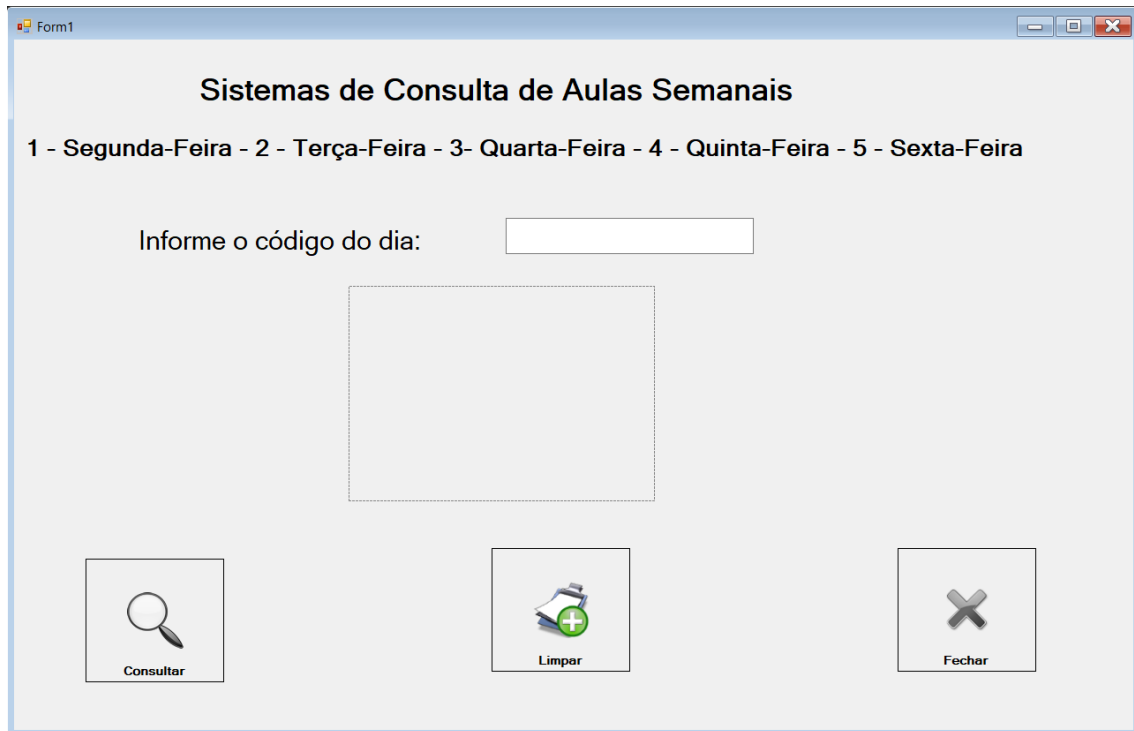
- Primeiro, libere a memória que está sendo usada pela imagem e, em seguida, limpe o gráfico. A coleta de lixo liberará a memória mais tarde se o gerenciamento de memória se tornar um problema.
- No exemplo o if verifica se a propriedade image é diferente != de null (se uma imagem foi atribuída).
- Em caso verdadeiro libera a memória e limpa o gráfico.

```
if (pictureBox1.Image != null)
{
    pictureBox1.Image.Dispose();
    pictureBox1.Image = null;
}
```



## Exemplo utilizando imagens na estrutura **switch/case**

O objetivo é mostrar no pictureBox (caixa de imagem) uma imagem relativa ao dia da semana, e em um MessageBox as matérias do dia e seus professores.

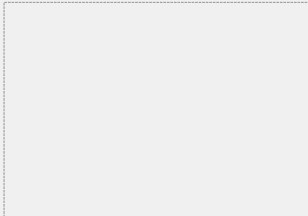


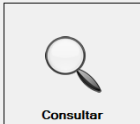
Form1


**Sistemas de Consulta de Aulas Semanais**


1 - Segunda-Feira - 2 - Terça-Feira - 3- Quarta-Feira - 4 - Quinta-Feira - 5 - Sexta-Feira

Informe o código do dia:



 Consultar

 Limpar

 Fechar

### Names:

TextBox txtDia;

PictureBox picImagem;

Button btnConsultar;

Button btnLimpar;

Button btnFechar;

```

private void btnFechar_Click(object sender, EventArgs e)
{
    MessageBox.Show("Encerrando o Sistema.....");
    Close();
}

private void btnConsultar_Click(object sender, EventArgs e)
{
    int dia;
    dia = Int32.Parse(txtDia.Text);

    switch (dia)
    {
        case 1:
        {
            picImagem.Image =
Image.FromFile("F:\\2022\\1 semestre\\2 Novotec\\DES\\segunda.jpg");
            picImagem.SizeMode = PictureBoxSizeMode.StretchImage;
            MessageBox.Show("ING - Elaine\\n PWEB - Rosiane\\n continue
colocando as matérias de segunda");

            break;
        }

        case 2:
        {
            picImagem.Image =
Image.FromFile("F:\\2022\\1 semestre\\2 Novotec\\DES\\terça.jpg");
            picImagem.SizeMode = PictureBoxSizeMode.StretchImage;
            MessageBox.Show("HIS - Claudio\\n ECO - Jaine\\n continue
colocando as matérias de terça");
            break;
        }

        case 3:
        {
            picImagem.Image =
Image.FromFile("F:\\2022\\1 semestre\\2 Novotec\\DES\\quarta.jpg");
            picImagem.SizeMode = PictureBoxSizeMode.StretchImage;
            MessageBox.Show("Continue colocando as matérias de
quarta");
            break;
        }

        case 4:
        {
            picImagem.Image =
Image.FromFile("F:\\2022\\1 semestre\\2 Novotec\\DES\\quinta.jpg");
            picImagem.SizeMode = PictureBoxSizeMode.StretchImage;

```

```

        MessageBox.Show("Continue colocando as matérias de
quinta");
        break;
    }

    case 5:
    {
        picImagem.Image =
Image.FromFile("F:\\2022\\1 semestre\\2Novotec\\DES\\sexta.jpeg");
        picImagem.SizeMode = PictureBoxSizeMode.StretchImage;
        MessageBox.Show("Continue colocando as matérias de sexta");
        break;
    }

    default:
    {
        MessageBox.Show("Opção Inválida");
        break;
    }

    }

}

private void btnLimpar_Click(object sender, EventArgs e)
{
    txtDia.Clear();
    picImagem.Image.Dispose();
    picImagem.Image = null;
}

private void txtDia_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar >= '0' && e.KeyChar <= '9' || e.KeyChar == 8)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
}

```

## ATIVIDADE



Uma nutricionista quer ajudar as pessoas a comerem alimentos que são da época.

Para isso você deve criar um sistema utilizando a estrutura switch/case que permita a digitação dos números equivalentes a todos os meses do ano:

1 – Janeiro

2 – Fevereiro

.

.

.

12 – Dezembro.

Como resultado você deve exibir 3 imagens: a primeira uma fruta, a segunda uma hortaliça e a terceira um legume que são da época.

Em um MessageBox exiba uma listagem com mais alimentos do respectivo mês.

Faça a programação do evento KeyPress para que aceite a digitação de caracteres numérico e a tecla backspace.

Aceite somente a digitação de dois caracteres para o mês

Insira um botão para limpar os dados consultados e um botão para encerrar o sistema.