

PROVA C# 1º BIM

SLIDES

Vetores

O que é um vetor?

- Coleção de valores de mesmo tipo acessíveis pelo índice (index) de inclusão.

Declaração: TIPO [] nome = new TIPO[TAMANHO];

```
1 double[] notas = new double[4];  
2 string [] alunos = new string[5] { "Ana", "João", "Lucas", "Raquel", "Tamires" };
```

FOREACH

Essa é a nova estrutura que o C# nos proporciona, seu propósito é percorrer coleções sem que haja a necessidade de acessar seus valores pelo índice!

```
1 string [] alunos = new string[5] { "Ana", "João", "Lucas", "Raquel", "Tamires" };  
2  
3 //AO INVÉS DISSO  
4 for(int i = 0; i < alunos.Length; i++)  
5 {  
6     Console.WriteLine(alunos[i]);  
7 }  
8  
9 //PODE-SE FAZER ISSO  
10 foreach(string aluno in alunos)  
11 {  
12     Console.WriteLine(aluno);  
13 }
```

Para praticar... Cálculo de médias

Sabe-se que no CTI a média das disciplinas é 6.

Crie uma aplicação que receba o nome do aluno, as notas dos quatro bimestres, calcule a média final, exiba o seu nome, a média e se ele foi reprovado ou aprovado.

Faça essa repetição enquanto o nome do aluno não for 0.

Ao final, calcular a média geral da turma e exibir.

USE VETORES.

EXERCICIO DESSE SLIDE:

```
int qtd = 0;  
double[] notas= new double[4];  
double soma=0;  
string? nome;
```

```

string? j;

do
{
    Console.Write("digite seu nome: ");
    nome = Console.ReadLine();

    if (nome != "")
    {
        double media = 0;
        for (int i = 0; i < 4; i++)
        {
            Console.Write("digite a nota do " + (i + 1) + " bimestre: ");
            j = Console.ReadLine();
            Double.TryParse(j, out notas[i]);
            media += notas[i];
        }
        media /= 4;

        Console.WriteLine("aluno: " + nome);
        Console.WriteLine("media: " + String.Format("{0:.#}", media));

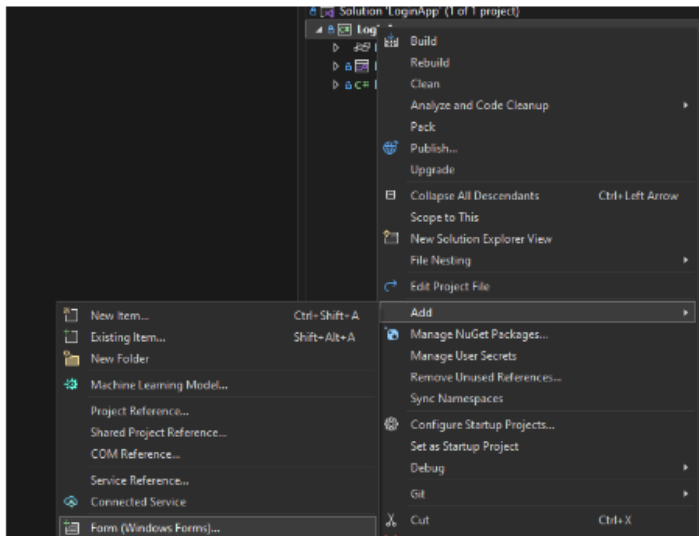
        if (media < 6)
        {
            Console.WriteLine("aluno reprovado");
        }
        else
        {
            Console.WriteLine("aluno aprovado");
        }

        soma += media;
        qtd++;
    }
} while (nome != "");

Console.WriteLine("media da turma: " + String.Format("{0:.#}", soma/qtd));

```

Aplicações multifarmulário



Clique com o botão direito no Solution Explorer e vá até “Add”, depois selecione “Form (Windows Forms)”

```
FrmPrincipal frm = new FrmPrincipal();  
frm.Show();
```

Essas duas linhas acima criam uma variável do tipo FrmPrincipal e, logo em seguida, com o método .Show(), faz com que ele apareça.

```
this.Hide();
```

Esse método oculta o formulário que o chama (por causa da palavra-chave this).

Menu: Componente MenuStrip.

EXERCICIO IMC

```
//programação orientada a eventos  
namespace imcApp  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private void Form1_Load(object sender,  
EventArgs e)  
        {  
        }  
    }  
}
```

```
private void label3_Click(object sender,  
EventArgs e)  
{  
}  
  
private void label2_Click(object sender,  
EventArgs e)  
{  
}  
  
private void  
numericUpDown1_ValueChanged(object  
sender, EventArgs e)  
{  
}
```

```

        return "obesidade grau III";
    }

    decimal Imc(decimal p, decimal h)
    {
        return p/(h*h);
    }

    string Classificacao(decimal imc)
    {
        double i = (double)imc;

        /*nn pode comparar decimal com
double então criamos um
double com o mesmo valor que imc
para comparar double
com double */

        if (i < 18.6)
            return "abaixo do peso.";

        if (i < 25)
            return "peso ideal.";

        if (i < 30)
            return "levemente acima do peso.";

        if (i < 35)
            return "obesidade grau I.";

        if (i < 40)
            return "obesidade grau II.";
    }
}

return "obesidade grau III";
}

private void button1_Click(object sender,
EventArgs e)
{
    decimal peso, altura; //tipo ponto
flutuante

    peso = numPeso.Value; //valor que
usuario digitou
    altura= numAltura.Value;

    if(altura == 0)
    {
        MessageBox.Show("bota uma altura
valida cacete");
    }
    else
    {
        decimal imc = Imc(peso, altura);

        lblcalculo.Text =
imc.ToString("0.0");//converte a variavel imc
para string e joga pra dentro do label
        lblclass.Text = Classificacao(imc);
    }
}
}
}

```

- Crie uma aplicação que, a partir de um menu, permita que o usuário escolha entre as seguintes equações físicas:
 - Velocidade média ($V = S/t$);
 - Aceleração ($a = V/t$);
 - Movimento Uniforme ($S = S_0 + V*t$);
 - Torricelli ($V = \sqrt{V_0^2 + 2*a*S}$);
 - Sair.
- Depois da escolha, receber como entrada os parâmetros da equação, calcular e retornar o valor;
- Voltar ao menu e repetir enquanto não se selecionar "Sair".
- **USE FUNÇÕES!!!!**

CODIGO ABAIXO:

```

int Menu()
{
    Console.WriteLine("=====
FisicaApp! =====");
    Console.WriteLine("1- veloc média");

    Console.WriteLine("2- aceleração");
    Console.WriteLine("3- mov uniforme");
    Console.WriteLine("4- Torricelli");
    Console.WriteLine("5- Sair");
}

```

```

    Console.WriteLine("escolha a opção
desejada:");
    int op = Int32.Parse(Console.ReadLine());

    return op;
}

```

```

double velocidademedia()
{
    double s, t;
    Console.WriteLine("espaço: ");
    s = Double.Parse(Console.ReadLine());
    Console.WriteLine("tempo: ");
    t = Double.Parse(Console.ReadLine());

    return s / t;
}

```

```

double aceleracao()
{
    double v, t;
    Console.WriteLine("Velocidade: ");
    v = Double.Parse(Console.ReadLine());
    Console.WriteLine("tempo: ");
    t = Double.Parse(Console.ReadLine());

    return v / t;
}

```

```

double movimentouniforme()
{
    double s0, v, t;

    Console.WriteLine("espaço inicial: ");
    s0 = Double.Parse(Console.ReadLine());
    Console.WriteLine("velocidade: ");
    v = Double.Parse(Console.ReadLine());
    Console.WriteLine("tempo: ");
    t = Double.Parse(Console.ReadLine());

    return s0 + v * t;
}

```

```

double torricelli()
{
    double v0, a, s;
    Console.WriteLine("velocidade inicial:");
    v0 = Double.Parse(Console.ReadLine());
    Console.WriteLine("aceleração:");
    a = Double.Parse(Console.ReadLine());
    Console.WriteLine("espaço: ");
    s = Double.Parse(Console.ReadLine());
}

```

```

    return Math.Sqrt(Math.Pow(v0, 2) + (2 * a *
s));
}

```

```

int escolha = 0;

```

```

do
{
    escolha = Menu();

    if(escolha == 1) //veloc media
    {
        double v = velocidademedia();

        Console.WriteLine("Velocidade média:
"+v);
    }
    else
    {
        if(escolha == 2)
        {
            double a = aceleracao();
            Console.WriteLine("aceleração: " + a);
        }
        else
        {
            if (escolha == 3)
            {
                double s = movimentouniforme();
                Console.WriteLine("espaço final: " +
s);
            }
            else
            {
                if (escolha == 4)
                {
                    double t = torricelli();
                    Console.WriteLine("velocidade
final: " + t);
                }

                else
                {
                    if(escolha ==5)
                    {
                        Console.WriteLine("tchau!");
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
}

```

```

}while(escolha !=0);

```

FISICA APP 2.0 BASICO

```

namespace FisicaApp_2._0____básico
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            btVM.Enabled = false;
            btA.Enabled = false;
        }

        private void btVM_Click(object sender,
            EventArgs e)
        {
            lblResult.Text = (num1.Value /
            num2.Value).ToString() + " m/s";
        }

        private void btA_Click(object sender,
            EventArgs e)
        {
            lblResult.Text = (num1.Value /
            num2.Value).ToString() + " m/s²";
        }
    }
}

```

```

        private void btMU_Click(object sender,
            EventArgs e)
        {
            lblResult.Text = (num1.Value +
            num2.Value * num3.Value).ToString() + " m";
        }

        private void btTorri_Click(object sender,
            EventArgs e)
        {
            lblResult.Text =
            Math.Sqrt((double)(num1.Value * num1.Value
            + (2 * num2.Value * num3.Value))).ToString() +
            " m/s";
        }

        private void num2_ValueChanged(object
            sender, EventArgs e)
        {
            btVM.Enabled = !(num2.Value == 0);
            btA.Enabled = !(num2.Value == 0);
        }
    }
}

```

FISICA APP 2.0 TOP

```

using System.Reflection;

namespace FisicaAPPTOP
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender,
            EventArgs e)
        {

```

```

        }

        private void
            rbVM_CheckedChanged(object sender,
            EventArgs e)
        {
            btCalc.Enabled = false;
            lblEntr1.Text = "Deslocamento:";
            lblEntr2.Text = "Tempo:";
            NUD_Entr1.Value = 0;
            NUD_Entr2.Value = 0;
            lblEntr3.Hide();
            NUD_Entr3.Hide();
            lblResult.Text = "";
        }
    }
}

```

```

        private void
rbAceleracao_CheckedChanged(object sender, EventArgs e)
    {
        btCalc.Enabled = false;
        lblEntr1.Text = "Velocidade:";
        lblEntr2.Text = "Tempo:";
        NUD_Entr1.Value = 0;
        NUD_Entr2.Value = 0;
        lblEntr3.Hide();
        NUD_Entr3.Hide();
        lblResult.Text = "";
    }

    private void
rbMU_CheckedChanged(object sender, EventArgs e)
    {
        btCalc.Enabled = true;
        lblEntr1.Text = "Espaço inicial:";
        lblEntr2.Text = "Velocidade:";
        lblEntr3.Show();
        lblEntr3.Text = "Tempo:";
        NUD_Entr1.Value = 0;
        NUD_Entr2.Value = 0;
        NUD_Entr3.Show();
        NUD_Entr3.Value = 0;
        lblResult.Text = "";
    }

    private void
rbTorricelli_CheckedChanged(object sender, EventArgs e)
    {
        btCalc.Enabled = true;
        lblEntr1.Text = "Velocidade inicial:";
        lblEntr2.Text = "Aceleração:";
        lblEntr3.Show();
        lblEntr3.Text = "Deslocamento:";
        NUD_Entr1.Value = 0;
        NUD_Entr2.Value = 0;
        NUD_Entr3.Show();
    }

```

```

        NUD_Entr3.Value = 0;
        lblResult.Text = "";
    }

    private void btCalc_Click(object sender, EventArgs e)
    {
        if (rbVM.Checked)
        {
            lblResult.Text = (NUD_Entr1.Value /
NUD_Entr2.Value).ToString() + " m/s";
        }
        else if (rbAceleracao.Checked)
        {
            lblResult.Text = (NUD_Entr1.Value /
NUD_Entr2.Value).ToString() + " m/s²";
        }
        else if (rbMU.Checked)
        {
            lblResult.Text = (NUD_Entr1.Value +
NUD_Entr2.Value *
NUD_Entr3.Value).ToString() + " m";
        }
        else
        {
            lblResult.Text =
Math.Sqrt((double)(NUD_Entr1.Value *
NUD_Entr1.Value + (2 * NUD_Entr2.Value *
NUD_Entr3.Value))).ToString() + " m/s";
        }
    }

    private void
NUD_Entr2_ValueChanged(object sender, EventArgs e)
    {
        {
            btCalc.Enabled =
!((rbAceleracao.Checked || rbVM.Checked) &&
NUD_Entr2.Value == 0);
        }
    }
}

```

EXERCICIO PAGINA DE LOGIN

Crie uma nova aplicação que funcionará como uma tela de login, deverá ter um campo de texto para o login, um campo de senha (MaskedTextBox), uma checkbox para exibir ou não a senha e um botão de entrar.

Quando clicar no botão, valide o usuário e senha, caso sejam "admin" e "ra"+seu RA (respectivamente), exiba uma MessageBox com "Acesso permitido", do contrário, mostre "Credenciais incorretas!"

Quando o form for carregado, também exibir em uma MessageBox "Bem-vindo ao sistema de login de teste!"

```
namespace loginEsenha
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void Form1_Load(object sender,
            EventArgs e)
            {
                MessageBox.Show("Bem-vindo ao
                sistema de login de teste!");
            }

            private void
            checkBox1_CheckedChanged(object sender,
            EventArgs e)
            {
                if (checkBox1.Checked)
                {
                    mskSenha.PasswordChar = '\0';
                }
                else
                {
                    mskSenha.PasswordChar = '*';
                }
            }

            private void
            mskSenha_MaskInputRejected(object sender,
            MaskInputRejectedEventArgs e)
            {
            }

            private void btLogar_Click(object sender,
            EventArgs e)
            {
                if (txtLogin.Text == "admin" &&
                mskSenha.Text == "ra2357097")
                {
                    MessageBox.Show("Acesso
                    permitido.");
                }
                else
                {
                    MessageBox.Show("Credenciais
                    incorretas!");
                }
            }

            private void txtLogin_TextChanged(object
            sender, EventArgs e)
            {
            }
        }
    }
}
```

EXERCICIOS LISTA

- 1) Faça um algoritmo que leia 10 salários. Depois de lidos e armazenados, mostre o maior valor. Utilize vetores.
- 2) Armazene num vetor de 5 posições o salário de 5 pessoas. Se o salário for menor q 1000 reais, forneça um aumento de 10% e sobrescreva o valor antigo. Ao final, mostre a lista de salários atualizada.
- 3) Faça um programa q leia n valores reais. Armazene estes valores num vetor. Ao final, imprima a média aritmética destes valores.
- 4) Faça um programa q pergunte ao usuário o numero de alunos a ser lido. O tamanho dos vetores será o numero informado pelo usuário. Armazene num vetor as notas G1 destes alunos; num outro vetor, armazene as notas G2 destes alunos. Ambas notas, G1 e G2, são informadas pelo usuário. Calcule a media aritmética destes alunos e armazene num terceiro vetor. Ao final, mostre as 3 notas dos alunos.
- 5) Faça um algoritmo q leia e armazene 5 valores inteiros em um vetor Vet1. Leia outros 5 valores inteiros e armazene num vetor Vet2. A partir destes valores lidos, mostre na tela:
 - a) a soma dos elementos de cada vetor, nas respectivas posições
 - b) a diferença dos elementos de cada vetor, nas respectivas posições
 - c) o produto dos elementos de cada vetor, nas respectivas posições
 - d) a divisão entre os elementos de cada vetor, nas respectivas posições

```
double [] salarios= new double [10];
double[] salarios2 = new double[5];
double maior = 0;
string? j;
string? n;
double soma = 0;

//exercicio 1
Console.WriteLine("====exercicio 1====\n");
for (int i = 0; i<salarios.Length; i++)
{
    Console.Write("digite um salario (posição "
+ (i+1) + "):");
    j=Console.ReadLine();

    Double.TryParse(j, out salarios[i]);

    if (salarios[i] > maior)
    {
        maior = salarios[i];
    }
}

Console.WriteLine("o maior dos salarios é: " +
maior);
```

//exercicio 2

```
Console.WriteLine("\n====exercicio
2====\n");
for (int m = 0; m < salarios2.Length; m++)
{
    Console.Write("digite seu salario (posição "
+ (m + 1) + "):");
    n = Console.ReadLine();

    Double.TryParse(n, out salarios2[m]);
```

```
if (salarios2[m]<1000)
{
    Console.WriteLine("\nvocê receberá um
aumento de 10%!!\n");
    salarios2[m] += (10.0 / 100.0) *
salarios2[m];
}
}
```

```
Console.WriteLine("\n=====lista
atualizada dos salários:=====");
foreach(double salario in salarios2)
{
    Console.WriteLine(salario);
}
```

//exercicio 3

```
Console.WriteLine("\n====exercicio
3====\n");
int qtd;
```

```
Console.Write("digite a quantidade de valores
que deseja inserir:");
qtd = Int32.Parse(Console.ReadLine());
```

```
double[] num = new double[qtd];
```

```
for(int a= 0; a < qtd; a++)
{
    Console.WriteLine("digite o valor
"+(a+1)+"");
    num[a]=
Double.Parse(Console.ReadLine());
    soma += num[a];
}
```

```
double media = soma / qtd;
```

```

Console.WriteLine("\n Media aritmetica: " +
String.Format("{0:##}", media));

//exercicio 4
Console.WriteLine("\n====exercicio
4====\n");
int qtdAlunos;

Console.Write("digite o número de alunos que
deseja:");
qtdAlunos = Int32.Parse(Console.ReadLine());

double[] g1 = new double[qtdAlunos];
double[] g2 = new double[qtdAlunos];

Console.WriteLine("\nG1:");
for (int i=0; i<qtdAlunos; i++)
{
    Console.WriteLine($"Indigite a nota g1 do
{i+1}º aluno: ");
    g1[i]= Double.Parse(Console.ReadLine());
}

Console.WriteLine("\nG2:");
for (int i = 0; i < qtdAlunos; i++)
{
    Console.WriteLine($"Indigite a nota g2 do
{i+1}º aluno: ");
    g2[i] = Double.Parse(Console.ReadLine());
}

double[] mediaArit = new double[qtdAlunos];

for(int i=0; i < qtdAlunos; i++)
{
    mediaArit[i]= (g1[i] + g2[i])/qtdAlunos;
}

Console.WriteLine("\n=====Notas
Finais=====");

for (int i=0;i<qtdAlunos; i++)
{
    Console.WriteLine($"Aluno {(i+1)}");
    Console.WriteLine($"G1= " + g1[i]);
    Console.WriteLine($"G2= " + g2[i]);
    Console.WriteLine($"Média: ({g1[i]} + {g2[i]})
/ {qtdAlunos}= " + String.Format("{0:##}",
mediaArit[i]));
}

```

//exercicio 5

```

Console.WriteLine("\n====exercicio
5====\n");
int[] vet1 = new int[5];
int[] vet2 = new int[5];

for (int c = 0; c < 5; c++)
{
    Console.Write("digite um valor para o
primeiro vetor(" + (c+1) + "): ");
    vet1[c]= Int32.Parse(Console.ReadLine());
}

for (int w = 0; w < 5; w++)
{
    Console.Write("digite um valor para o
segundo vetor(" + (w+1) + "): ");
    vet2[w] = Int32.Parse(Console.ReadLine());
}

for(int i=0; i < 5; i++)
{
    Console.WriteLine("\nsoma dos
elementos:");
    Console.WriteLine($"posição {i + 1}: {vet1[i]}
+ {vet2[i]}=" + (vet1[i] + vet2[i]));
}

for (int i = 0; i < 5; i++)
{
    Console.WriteLine("\nsubtração dos
elementos:");
    Console.WriteLine($"posição {i+1}: {vet1[i]} -
{vet2[i]}=" + (vet1[i] - vet2[i]));
}

for (int i = 0; i < 5; i++)
{
    Console.WriteLine("\nproduto dos
elementos:");
    Console.WriteLine($"posição {i + 1}: {vet1[i]}
* {vet2[i]}=" + (vet1[i] * vet2[i]));
}

for (int i = 0; i < 5; i++)
{
    Console.WriteLine("\ndivisão dos
elementos:");
    if (vet2[i] != 0)
    {
        Console.WriteLine($"posição {i+1}:
{vet1[i]} / {vet2[i]}=" + ((double)vet1[i] / vet2[i]));
    }
}

```

```
else
{
    Console.WriteLine("não foi possível
realizar a divisão pois o número do segundo
vetor é 0.");
}
}
```