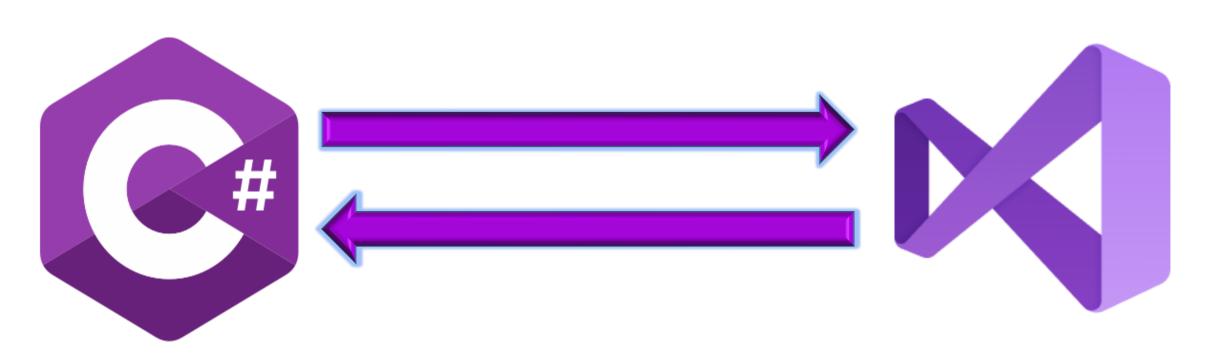
Linguagem de Programação Visual 1





ATIVIDADE LOTERIAS



Vamos implementar geradores de números para as seguintes loterias:















TRABALHANDO COM TABELAS Vamos iniciar pela Lotofácil

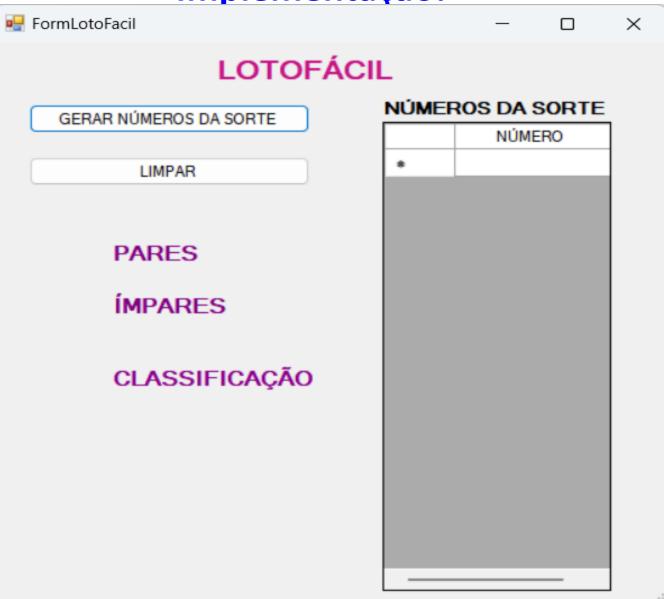


COMO FUNCIONA?

A Lotofácil é, como o próprio nome diz, fácil de apostar e principalmente de ganhar. Você marca entre 15 e 20 números, dentre os 25 disponíveis no volante, e fatura prêmio se acertar 11, 12, 13, 14 ou 15 números.



Implementação:







TRABALHANDO COM TABELAS O que será implementado:



- ✓ Gerador de números em ordem crescente;
- ✓ Estatística sobre a quantidade de números pares e ímpares
- ✓ Classificação da qualidade dos números gerados:





A estatísticas para fundamentar a implementação

•	-
Todos os concursos	Apenas os últimos 10 concursos
3 pares e 12 ímpares 3 vezes (0,11%)	3 pares e 12 ímpares 0 vezes (0,00%)
4 par e 11 ímpares 30 vezes (1,06%)	4 par e 11 ímpares 0 vezes (0,00%)
5 pares e 10 ímpares 198 vezes (6,96%)	5 pares e 10 ímpares 0 vezes (0,00%)
6 pares e 9 ímpares 594 vezes (20,89%)	6 pares e 9 ímpares 2 vezes (20,00%)
7 pares e 8 ímpares 879 vezes (30,92%)	7 pares e 8 ímpares 2 vezes (20,00%)
8 pares e 7 ímpares 725 vezes (25,50%)	8 pares e 7 ímpares 2 vezes (20,00%)
9 pares e 6 ímpares	9 pares e 6 ímpares
327 vezes (11,50%)	4 vezes (40,00%)
10 pares e 5 ímpares	10 pares e 5 ímpares
77 vezes (2,71%)	0 vezes (0,00%)
11 pares e 4 ímpares	11 pares e 4 ímpares
10 vezes (0,35%)	0 vezes (0,00%)





A estatísticas para fundamentar a implementação

Todos os concursos

3 pares e 12 ímpares 3 vezes (0,11%)

4 par e 11 ímpares 30 vezes (1,06%)

5 pares e 10 impares 198 vezes **(6,96%)**

6 pares e 9 impares 594 vezes (20,89%)

7 pares e 8 ímpares 879 vezes (30,92%)

8 pares e 7 impares 725 vezes (25,50%)

9 pares e 6 impares 327 vezes (11,50%)

10 pares e 5 impares 77 vezes (2,71%)

11 pares e 4 impares 10 vezes **(0,35%)**

7 pares e 8 ímpares => "MUITO ALTO!";

8 pares e 7 ímpares => "ALTO!";

6 pares e 9 ímpares => "MÉDIO!";

9 pares e 6 impares => "BAIXO!";

Outras combinações => "MUITO ALTO!";





Códigos: declaração da lista

```
public partial class FormLotoFacil : Form
    public List<int> NumerosDaSorte= new List<int>();
    public FormLotoFacil()
        InitializeComponent();
```





Códigos: Função de comparação (estatística)

```
private void Comparacao(int par, int impar)
    lbPar.Text = "PARES: " + par;
    lbImpar.Text = "IMPARES: " + impar;
    if (impar == 8 && par == 7)
        lbClass.Text = "MUITO ALTO!";
        lbClass.ForeColor = Color.Green;
    if (impar == 7 && par == 8) { lbClass.Text = "ALTO!"; lbClass.ForeColor = Color.Green; }
    if (impar == 9 && par == 6) { lbClass.Text = "MÉDIO!"; lbClass.ForeColor = Color.Orange; }
    if (impar == 6 && par == 9) { lbClass.Text = "BAIXO!"; lbClass.ForeColor = Color.OrangeRed; }
    if (impar <= 5 && par >= 10) { lbClass.Text = "MUITO BAIXO!"; lbClass.ForeColor = Color.Red; }
    if (impar >= 10 && par <= 5) { lbClass.Text = "MUITO BAIXO!"; lbClass.ForeColor = Color.Red; }
```





Códigos: Gerar números

```
public void GerarNumeros()
   int numero =0;
   int cont = 0;
    int qtdPar = 0;//quantidade de números pares
    int qtdImpar = 0;//quantidade de números impares
    Random radNum = new Random(); //objeto para gerar números aleatórios
    NumerosDaSorte.Clear();
   while (cont < 15)
       numero = radNum.Next(1, 26); //números aleatórios
        if (!NumerosDaSorte.Contains(numero)) //Contains => verifica se o número está na lista
           NumerosDaSorte.Add(numero);
           if (numero % 2 == 0) qtdPar++;
           if (numero % 2 == 1) qtdImpar++;
           cont++;
        }//fim do IF
    }//fim do laço
    NumerosDaSorte = NumerosDaSorte.OrderBy(num => num).ToList(); //colocar em ordem crescente
    Comparacao(qtdPar, qtdImpar);
   dtvNumeros.DataSource = NumerosDaSorte.Select(Numeros => new { Numero = Numeros }).ToList();
    //colocar os números na GRID (Tabela)
}//fim da função
```





Códigos: Gerar números

```
public void GerarNumeros()
   int numero =0;
    int cont = 0;
    int qtdPar = 0;//quantidade de números pares
    int qtdImpar = 0;//quantidade de números impares
    Random radNum = new Random(); //objeto para gerar números aleatórios
    NumerosDaSorte.Clear();
```





```
Códigos: Gerar números = new Random(); //objeto para gerar numeros al
    NumerosDaSorte.Clear();
    while (cont < 15)
        numero = radNum.Next(1, 26); //números aleatórios
        if (!NumerosDaSorte.Contains(numero)) //Contains => verifica se o número está na lista
            NumerosDaSorte.Add(numero);
            if (numero % 2 == 0) qtdPar++;
            if (numero % 2 == 1) qtdImpar++;
            cont++;
        }//fim do IF
    }//fim do laço
    NumerosDaSorte = NumerosDaSorte.OrderBy(num => num).ToList(); //colocar em ordem crescente
    Comparacao(qtdPar, qtdImpar);
    dtvNumeros.DataSource = NumerosDaSorte.Select(Numeros => new { Numero = Numeros }).ToList();
    //colocar os números na GRID (Tabela)
}//fim da função
```





Códigos: Limpar tabela

```
public void LimparTabela()
{
   NumerosDaSorte.Clear();
   dtvNumeros.DataSource = NumerosDaSorte.Select(Numeros => new { Numero = Numeros }).ToList();
}
```





Códigos dos botões:

```
private void btGerar_Click(object sender, EventArgs e)
    GerarNumeros();
private void btLimpar_Click(object sender, EventArgs e)
   LimparTabela();
    lbClass.Text = "CLASSIFICAÇÃO";
    lbPar.Text = "PARES";
    lbImpar.Text = "IMPARES";
```



ATIVIDADE AVALIATIVA Exercício => Grupo de 2(duas) pessoas



- 1 Implementar Mega Sena com estatísticas
- 2 Implementar Lotomania com estatísticas
- 3 Implementar Quina com estatísticas
- 4 Implementar Dia de Sorte com estatísticas
- 5 Implementar Super Sete com estatísticas