

Desenvolvimento com Java

Professor: Orlando Fonseca



Kaike Torres da silva

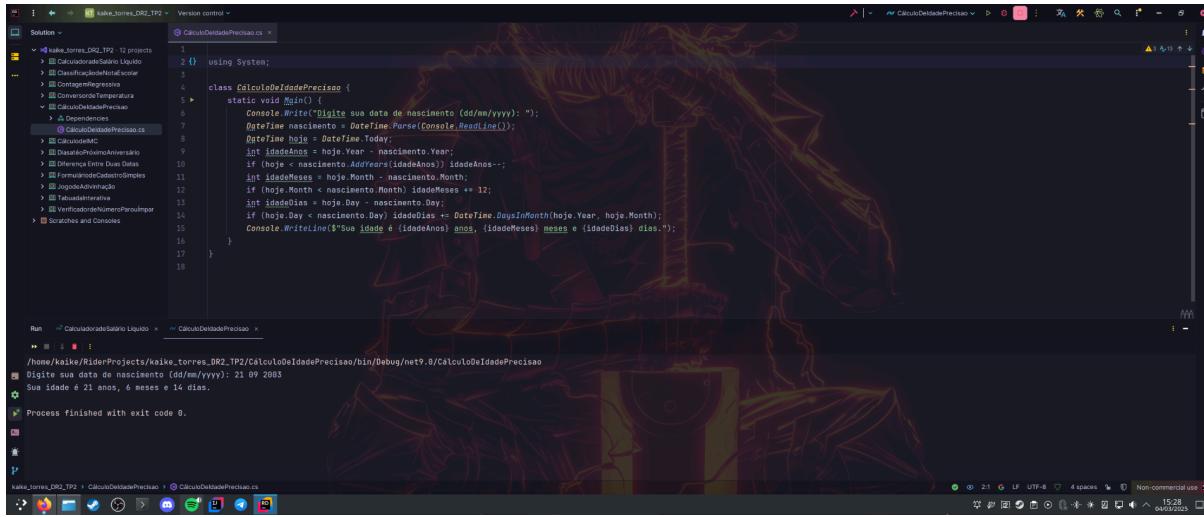
04.03.2025

Análise e Desenvolvimento de Sistemas (ADS)

INTRODUÇÃO

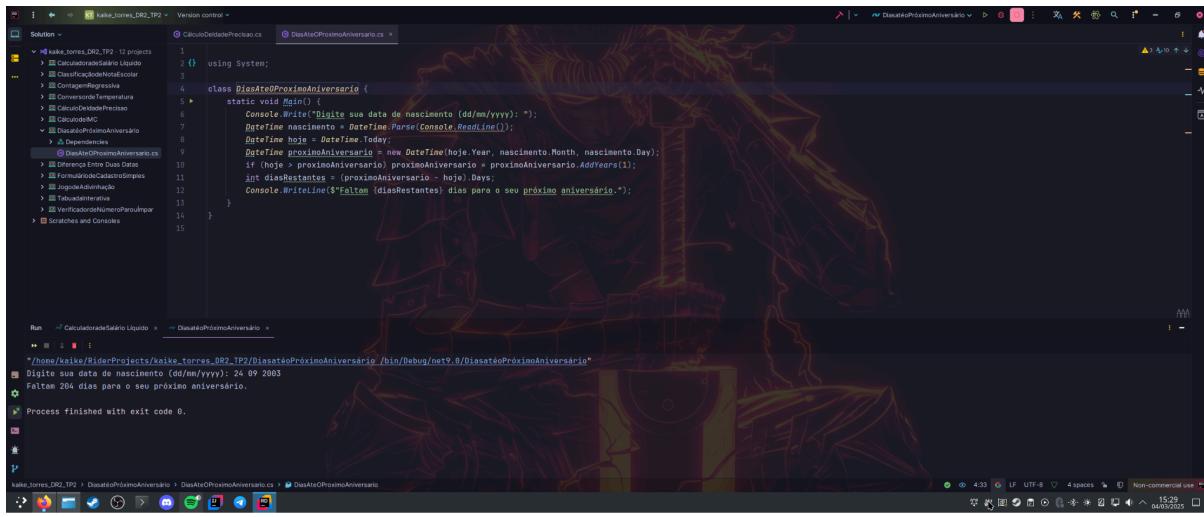
Este relatório apresenta a solução de um conjunto de exercícios desenvolvidos em C#, abrangendo conceitos fundamentais da linguagem e boas práticas de programação. O objetivo principal é aplicar estruturas de controle de fluxo, manipulação de datas, entrada de usuário e cálculos matemáticos de maneira eficiente e organizada.

Exercício 1: Cálculo de Idade Precisa



```
1 using System;
2
3 class CalculoDeIdadePreciso {
4     static void Main() {
5         Console.WriteLine("Digite sua data de nascimento (dd/mm/yyyy): ");
6         DateTime nascimento = DateTime.Parse(Console.ReadLine());
7         DateTime hoje = DateTime.Today;
8         int idadeAnos = hoje.Year - nascimento.Year;
9         if (hoje.Month < nascimento.Month) idadeAnos--;
10        int idadeMeses = hoje.Month - nascimento.Month;
11        if (idadeMeses < 0) idadeMeses += 12;
12        int idadeDias = hoje.Day - nascimento.Day;
13        if (hoje.Day < nascimento.Day) idadeDias -= DateTime.DaysInMonth(hoje.Year, hoje.Month);
14        Console.WriteLine($"Sua idade é {idadeAnos} anos, {idadeMeses} meses e {idadeDias} dias.");
15    }
16 }
17
18
```

Exercício 2: Dias até o Próximo Aniversário



```
1 using System;
2
3 class DiasAtePróximoAniversario {
4     static void Main() {
5         Console.WriteLine("Digite sua data de nascimento (dd/mm/yyyy): ");
6         DateTime nascimento = DateTime.Parse(Console.ReadLine());
7         DateTime hoje = DateTime.Today;
8         DateTime proximoAniversario = new DateTime(hoje.Year, nascimento.Month, nascimento.Day);
9         if (hoje > proximoAniversario) proximoAniversario = proximoAniversario.AddYears(1);
10        int diasRestantes = (proximoAniversario - hoje).Days;
11        Console.WriteLine($"Faltam {diasRestantes} dias para o seu próximo aniversário.");
12    }
13 }
14
15
```

Exercício 3: Diferença Entre Duas Data

The screenshot shows the Visual Studio Code interface with the following details:

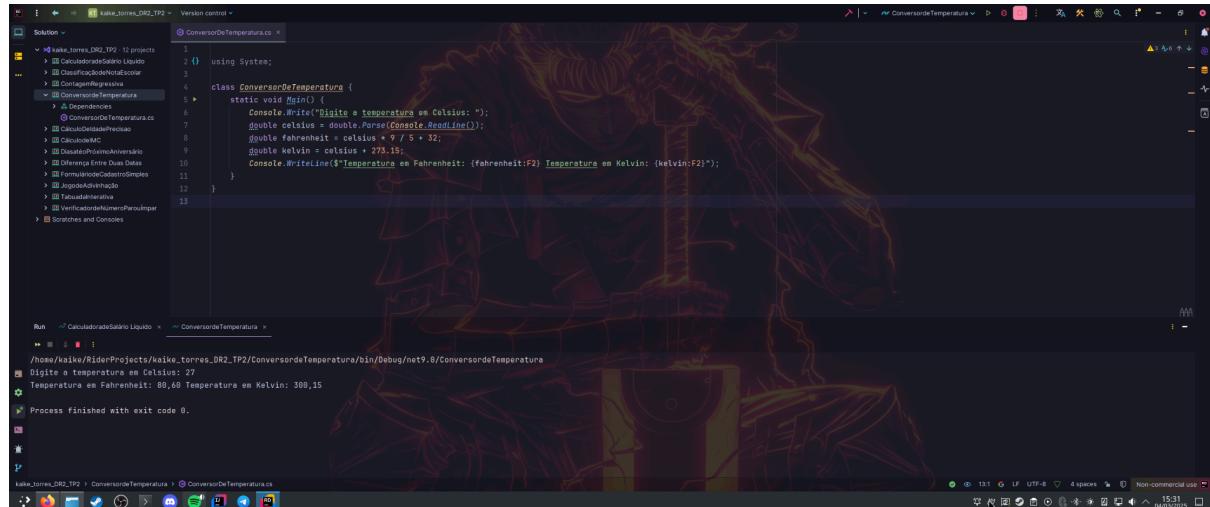
- Solution Explorer:** Shows 12 projects under the solution "kaile_torres_DR2_TP2". One project is expanded, showing files like CálculoDeIdadePessoas.cs, DatasAtéOPróximoAniversário.cs, and DiferençaEntreDatas.cs.
- Code Editor:** Displays the code for "DiferençaEntreDatas.cs". The code reads two dates from the console, calculates the difference in days, and then converts it into years, months, and days.
- Terminal:** Shows the output of the program running. It prompts for two dates, calculates the difference, and prints it as "A diferença entre as duas datas é de 11 anos, 5 meses e 22 dias."
- Status Bar:** Shows the current time as 15:10 and the date as 04/03/2025.

Exercício 4: Formulário de Cadastro Simples

The screenshot shows the Visual Studio Code interface with the following details:

- Solution Explorer:** Shows 12 projects under the solution "kaile_torres_DR2_TP2". One project is expanded, showing files like CálculoDeIdadePessoas.cs, DatasAtéOPróximoAniversário.cs, and FormulárioDeCadastroSimples.cs.
- Code Editor:** Displays the code for "FormulárioDeCadastroSimples.cs". The code reads user input for name, idade, telefone, and email.
- Terminal:** Shows the output of the program running. It prompts for user information and prints a summary line: "Nome: kaile_torres Idade: 21 Telefone: 96737234 E-mail: kailetorres@gmail.com".
- Status Bar:** Shows the current time as 15:10 and the date as 04/03/2025.

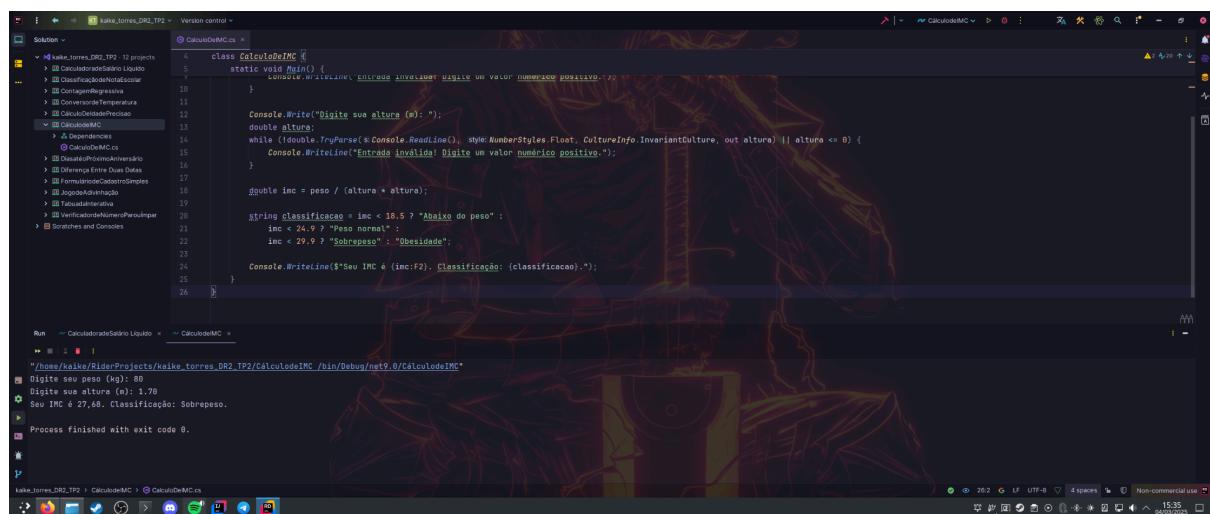
Exercício 5: Conversor de Temperatura



A screenshot of the Visual Studio IDE. The solution 'kaika_torres_DR2_TP2' contains 12 projects. The 'ConversorDeTemperatura.cs' file is open, showing C# code to convert Celsius to Fahrenheit and Kelvin. The output window shows the execution of the program, inputting 80.68 and getting outputs 176.8 and 300.15. The status bar at the bottom right shows '15:31' and 'Non-commercial use'.

```
1  using System;
2
3  class ConversorDeTemperatura {
4      static void Main() {
5          Console.WriteLine("Digite a temperatura em Celsius:");
6          double celsius = double.Parse(Console.ReadLine());
7          double fahrenheit = celsius * 9 / 5 + 32;
8          double kelvin = celsius + 273.15;
9          Console.WriteLine($"Temperatura em Fahrenheit: {fahrenheit:F2} Temperatura em Kelvin: {kelvin:F2}");
10     }
11 }
12
13
```

Exercício 6: Cálculo de IMC



A screenshot of the Visual Studio IDE. The solution 'kaika_torres_DR2_TP2' contains 12 projects. The 'CalculoDeIMC.cs' file is open, showing C# code to calculate BMI and classify it as Underweight, Normal, or Overweight. The output window shows the execution of the program, inputting 80 and 1.78, and getting an output of 27.68 and the classification 'Sobrepeso'. The status bar at the bottom right shows '15:39' and 'Non-commercial use'.

```
1  class CalculoDeIMC {
2      static void Main() {
3          Console.WriteLine("Entre com altura (m):");
4          double altura;
5          while (!double.TryParse(Console.ReadLine(), style: NumberStyles.Float, CultureInfo.InvariantCulture, out altura) || altura <= 0) {
6              Console.WriteLine("Entrada inválida! Digite um valor numérico positivo.");
7          }
8
9          double imc = peso / (altura * altura);
10
11         string classificacao = imc < 18.5 ? "Abaixo do peso";
12         imc < 24.9 ? "Peso normal";
13         imc > 29.9 ? "Sobrepeso";
14
15         Console.WriteLine($"Seu IMC é {imc:F2}. Classificação: {classificacao}.");
16     }
17
18 }
```

Exercício 7: Verificador de Número Par ou Ímpar

A screenshot of a dark-themed code editor (Visual Studio) showing a C# file named `VerificadorDeNumeroParOuImpar.cs`. The code checks if a user-specified integer is even or odd:

```
1  using System;
2
3  class VerificadorDeNumeroParOuImpar {
4      static void Main() {
5          Console.WriteLine("Digite um número inteiro:");
6          int numero = int.Parse(Console.ReadLine());
7          string resultado = numero % 2 == 0 ? "par" : "ímpar";
8          Console.WriteLine($"{numero} é {resultado}");
9      }
10 }
11
12 }
```

The terminal window shows the application running and prompting for input. The status bar at the bottom right indicates the date and time.

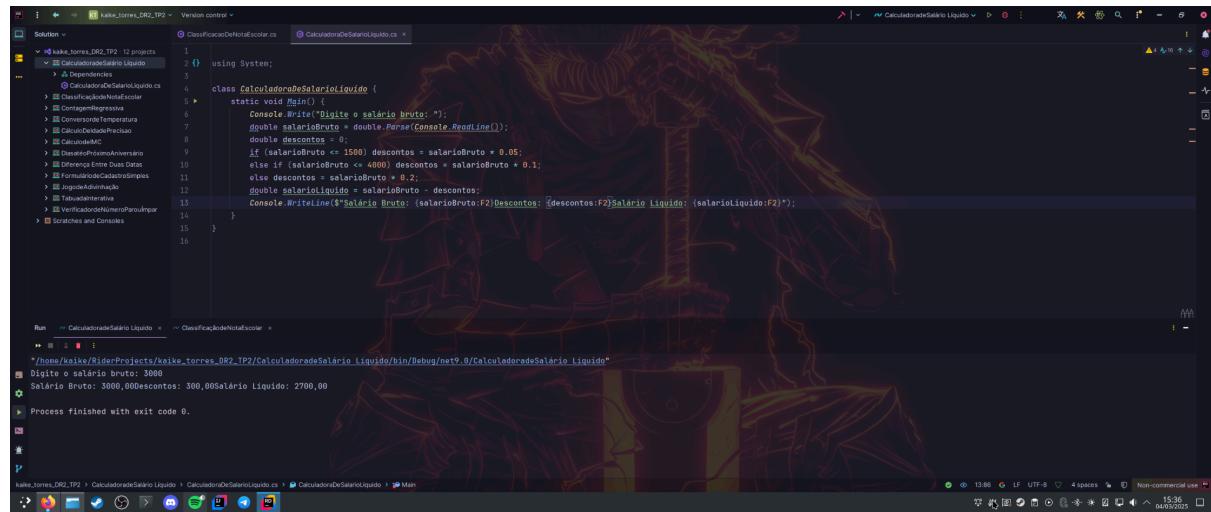
Exercício 8: Classificação de Nota Escolar

A screenshot of a dark-themed code editor (Visual Studio) showing a C# file named `ClassificacaoDeNotaEscolar.cs`. The code classifies a school grade (0-10) into four categories:

```
1  using System;
2
3  class ClassificacaoDeNotaEscolar {
4      static void Main() {
5          Console.WriteLine("Digite a nota (0-10):");
6          double nota = double.Parse(Console.ReadLine());
7          string classificacao = nota < 4 ? "Insuficiente";
8          nota < 7 ? "Regular";
9          nota < 9 ? "Bom";
10         "Excelente";
11
12     }
13
14 }
```

The terminal window shows the application running and prompting for input. The status bar at the bottom right indicates the date and time.

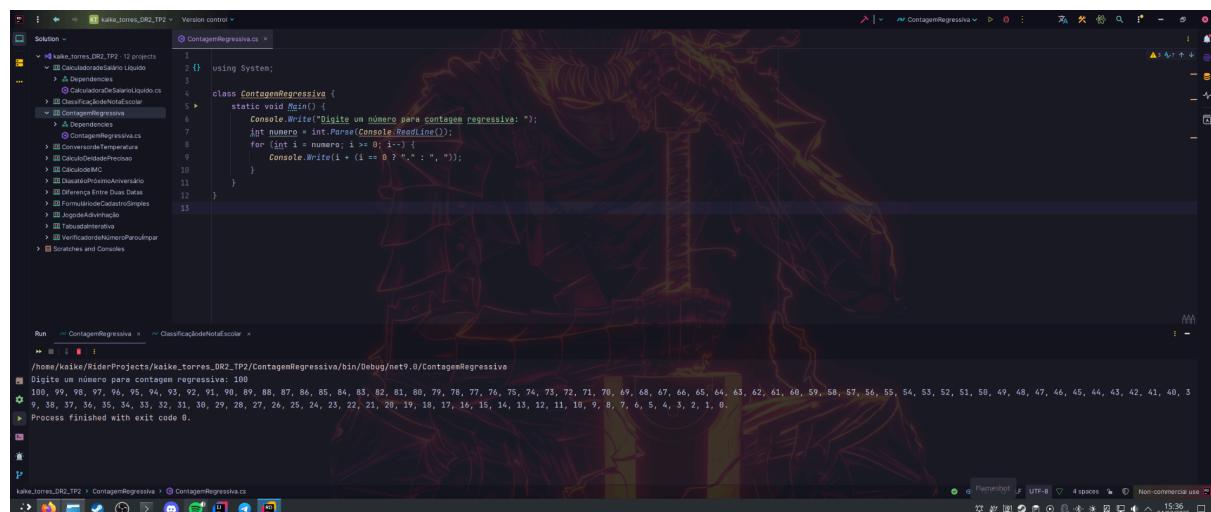
Exercício 9: Calculadora de Salário Líquido



```
using System;

class CalculadoraDeSalarioLiquido {
    static void Main() {
        Console.WriteLine("Digite o salário bruto:");
        double salarioBruto = double.Parse(Console.ReadLine());
        double descontos = 0;
        if (salarioBruto <= 1500) descontos = salarioBruto * 0.05;
        else if (salarioBruto <= 4000) descontos = salarioBruto * 0.1;
        else descontos = salarioBruto * 0.2;
        double salarioLiquido = salarioBruto - descontos;
        Console.WriteLine($"Salário Bruto: {salarioBruto:F2} Descontos: {descontos:F2} Salário Líquido: {salarioLiquido:F2}");
    }
}
```

Exercício 10: Contagem Regressiva



```
using System;

class ContagemRegressiva {
    static void Main() {
        Console.WriteLine("Digite um número para contagem regressiva:");
        int numero = int.Parse(Console.ReadLine());
        for (int i = numero; i >= 0; i--) {
            Console.WriteLine(i + i == 0 ? "" : " ");
        }
    }
}
```

Exercício 11: Tabuada Interativa

The screenshot shows the Visual Studio Code interface with the file `TabuadaInterativa.cs` open. The code defines a class `TabuadaInterativa` with a static `Main` method. It prompts the user to enter a number and then prints the multiplication table for that number from 1 to 10. The output window shows the results for the number 4.

```
1 using System;
2 
3 class TabuadaInterativa {
4     static void Main() {
5         Console.WriteLine("Digite um número para ver sua tabuada:");
6         int numero = int.Parse(Console.ReadLine());
7         for (int i = 1; i <= 10; i++) {
8             Console.WriteLine($"{numero} x {i} = {numero * i}");
9         }
10    }
11 }
12 
```

```
Process finished with exit code 0.
```

Exercício 12: Jogo de Adivinhação

The screenshot shows the Visual Studio Code interface with the file `JogoDeAdivinhação.cs` open. The code defines a class `jogo_de_adivinhação` with a static `Main` method. It generates a random secret number between 1 and 100 and asks the user to guess it. The program provides feedback if the guess is higher or lower than the secret number. The output window shows a sample session where the user guesses 50, 20, 40, 55, 53, 54, and finally 54 correctly.

```
1 using System;
2 
3 class jogo_de_adivinhação {
4     static void Main() {
5         Random random = new Random();
6         int numeroSecreto = random.Next(1, 101);
7         int palpite;
8         string resposta;
9         Console.WriteLine("Tente adivinhar o número secreto entre 1 e 100!");
10        while (palpite != numeroSecreto) {
11            Console.WriteLine("Digite seu palpite:");
12            palpite = int.Parse(Console.ReadLine());
13            if (palpite < numeroSecreto) Console.WriteLine("Maior!");
14            else if (palpite > numeroSecreto) Console.WriteLine("Menor!");
15        }
16        Console.WriteLine("Parabéns! Você acertou o número secreto.");
17    }
18 }
```

```
Process finished with exit code 0.
```

Conclusão

A resolução destes exercícios permitiu a aplicação prática de conceitos essenciais de programação em C#, consolidando o entendimento sobre manipulação de dados, entrada do usuário e lógica de controle de fluxo. Cada atividade desenvolvida contribuiu para o aprimoramento das habilidades necessárias para a escrita de códigos eficientes e bem estruturados.