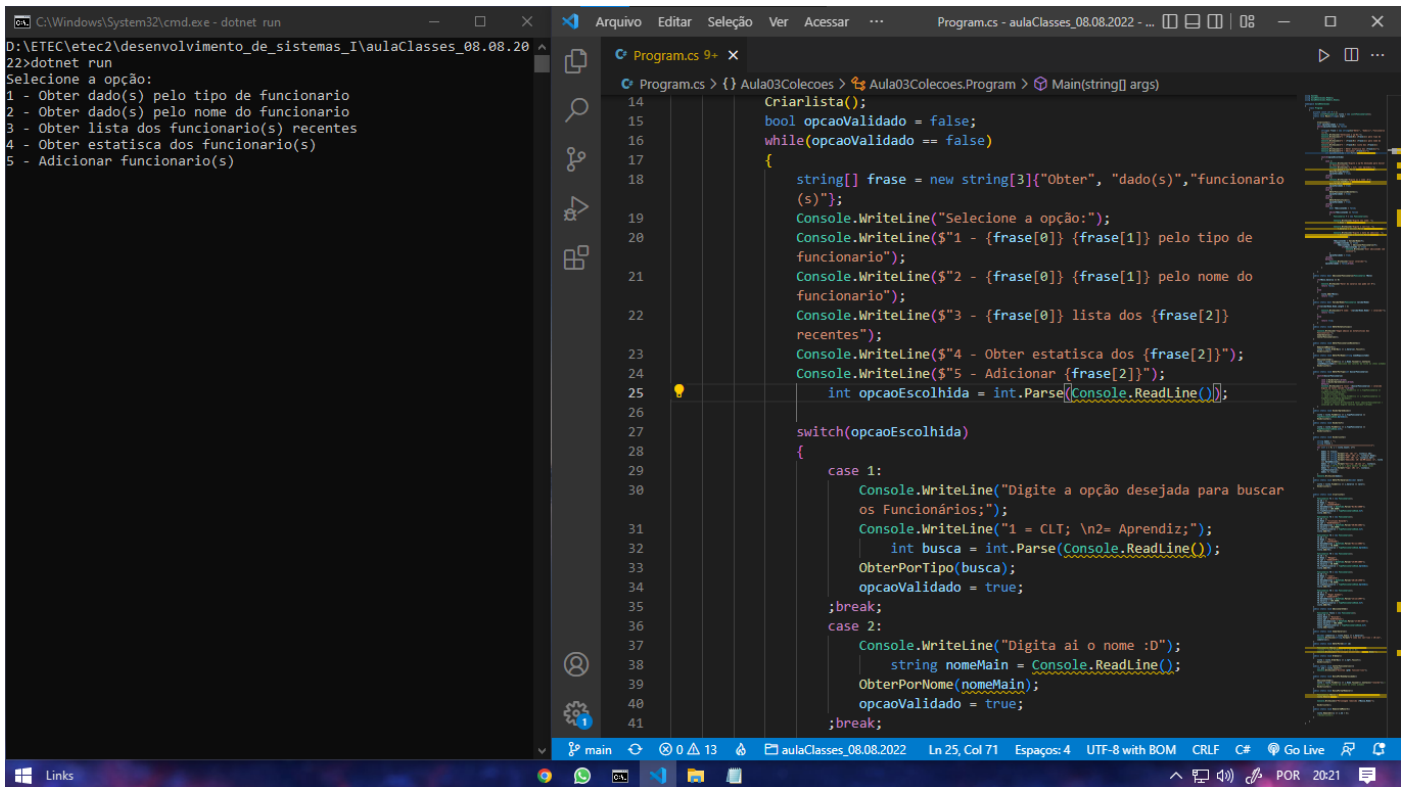


## 1) Exibir menu inicial.



The screenshot shows a Visual Studio Code editor with a C# console application. The left pane displays the command prompt output, and the right pane shows the source code in `Program.cs`.

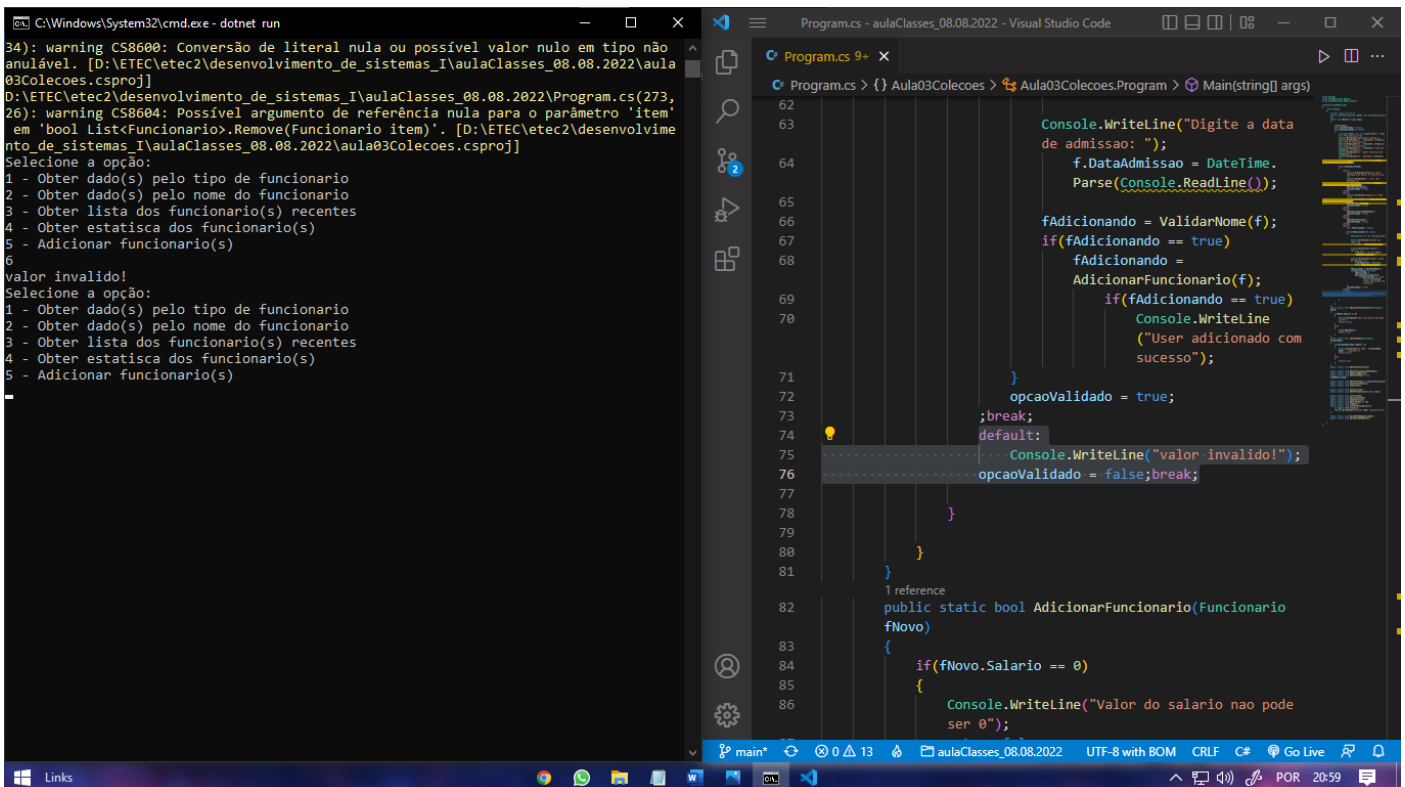
**Command Prompt Output:**

```
C:\Windows\System32\cmd.exe - dotnet run
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
```

**Source Code (Program.cs):**

```
14  Criarlista();
15  bool opcaoValidado = false;
16  while(opcaoValidado == false)
17  {
18      string[] frase = new string[3]{ "Obter", "dado(s)", "funcionario"
19      (s) };
20      Console.WriteLine("Selecione a opção:");
21      Console.WriteLine($"1 - {frase[0]} {frase[1]} pelo tipo de
22      funcionario");
23      Console.WriteLine($"2 - {frase[0]} {frase[1]} pelo nome do
24      funcionario");
25      Console.WriteLine($"3 - {frase[0]} lista dos {frase[2]}
26      recentes");
27      Console.WriteLine($"4 - Obter estatística dos {frase[2]}");
28      Console.WriteLine($"5 - Adicionar {frase[2]}");
29      int opcaoEscolhida = int.Parse(Console.ReadLine());
30
31      switch(opcaoEscolhida)
32      {
33          case 1:
34              Console.WriteLine("Digite a opção desejada para buscar
35              os Funcionários;");
36              Console.WriteLine("1 = CLT; \n2= Aprendiz;");
37              int busca = int.Parse(Console.ReadLine());
38              ObterPorTipo(busca);
39              opcaoValidado = true;
40              ;break;
41          case 2:
42              Console.WriteLine("Digite ai o nome :D");
43              string nomeMain = Console.ReadLine();
44              ObterPorNome(nomeMain);
45              opcaoValidado = true;
46              ;break;
```

## 2) Menu apresentando erro.



The screenshot shows the same Visual Studio Code editor, but now the command prompt displays an error message. The source code in `Program.cs` is also visible.

**Command Prompt Output:**

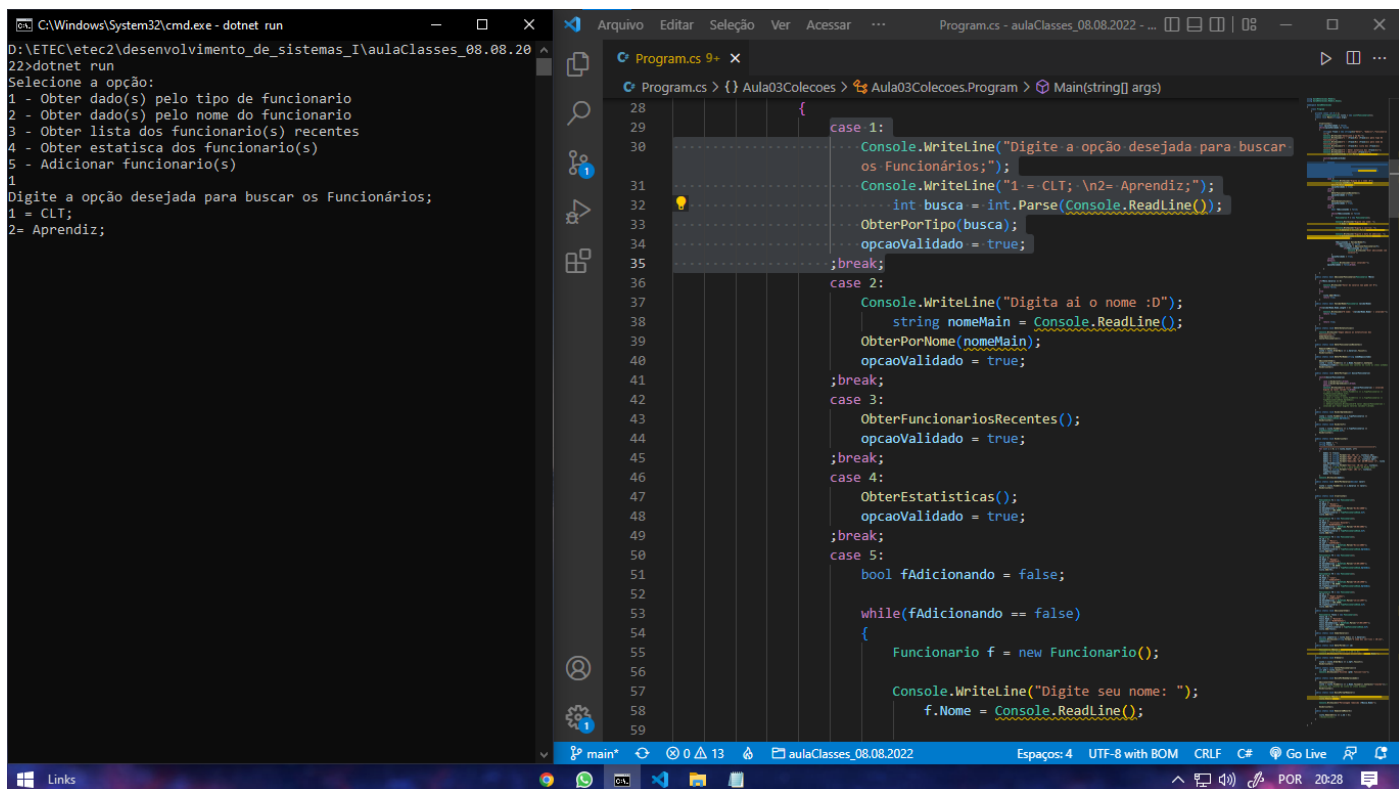
```
34): warning CS8600: Conversão de literal nula ou possível valor nulo em tipo não
anulável. [D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022\aula
03Colecoes.csproj]
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022\Program.cs(273,
26): warning CS8604: Possível argumento de referência nula para o parâmetro 'item'
em 'bool List<Funcionario>.Remove(Funcionario item)'. [D:\ETEC\etec2\desenvolvi
mento_de_sistemas_I\aulaClasses_08.08.2022\aula03Colecoes.csproj]
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
6
valor invalido!
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)

```

**Source Code (Program.cs):**

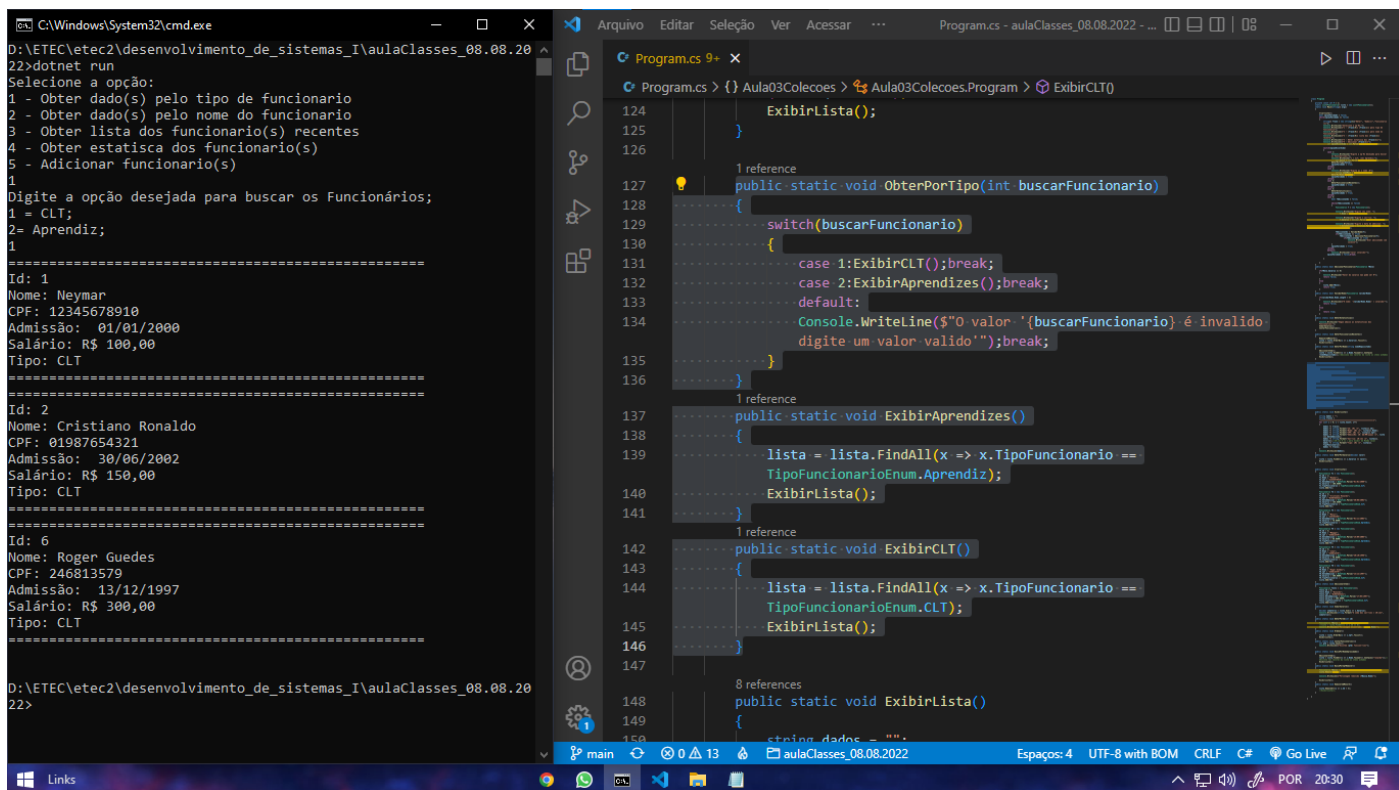
```
62  Console.WriteLine("Digite a data
63  de admissao: ");
64  f.DataAdmissao = DateTime.
65  Parse(Console.ReadLine());
66
67  fAdicionando = ValidarNome(f);
68  if(fAdicionando == true)
69  {
70      fAdicionando =
71      AdicionarFuncionario(f);
72      if(fAdicionando == true)
73      {
74          Console.WriteLine
75          ("User adicionado com
76          sucesso");
77      }
78      opcaoValidado = true;
79      ;break;
80      default:
81          Console.WriteLine("valor invalido!");
82          opcaoValidado = false; break;
83      }
84  }
85  }
86  }
87  }
88  }
89  }
90  }
91  }
92  }
93  }
94  }
95  }
96  }
97  }
98  }
99  }
100 }
```

### 3) Opção 1 selecionada.



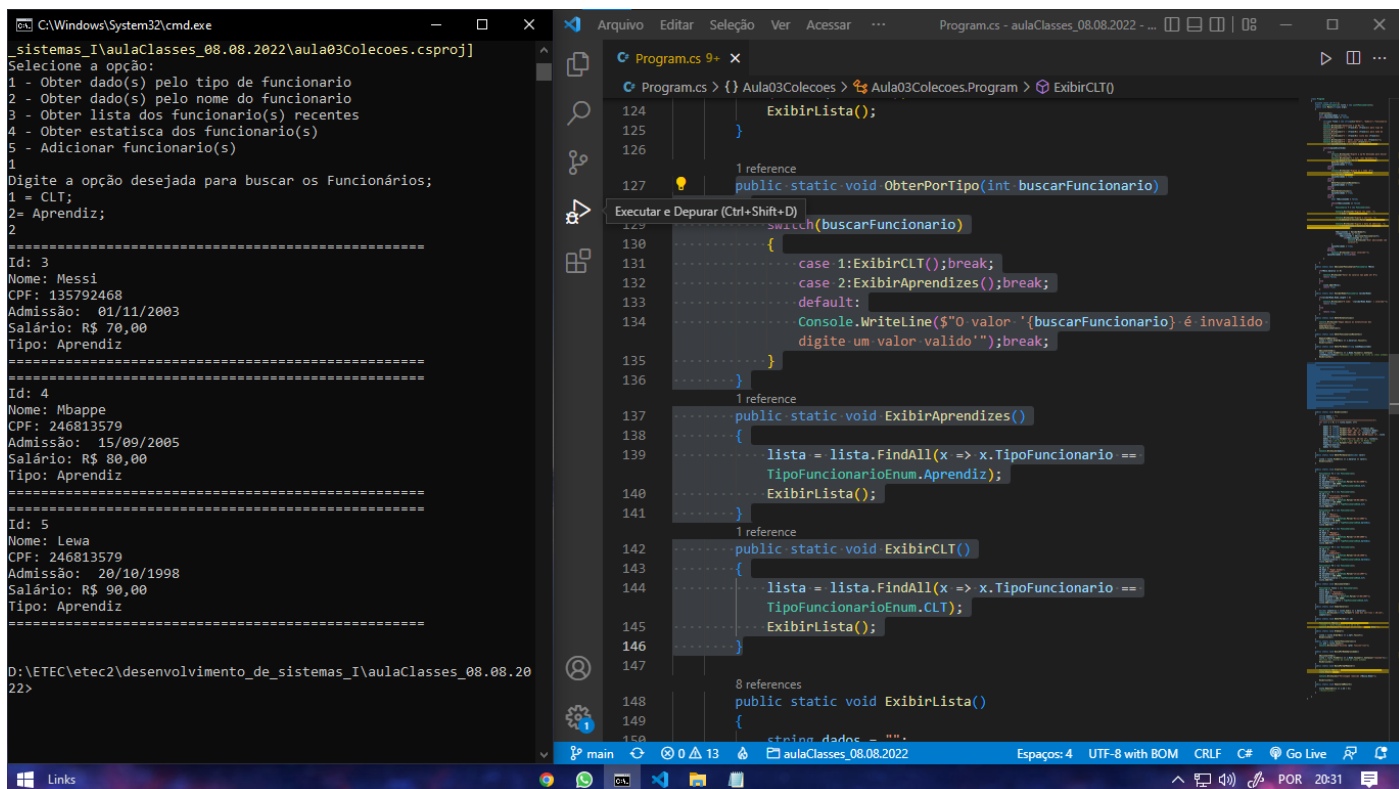
```
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
1
Digite a opção desejada para buscar os Funcionários;
1 = CLT;
2= Aprendiz;
1
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
Program.cs 9+ x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > Main(string[] args)
{
    case 1:
        Console.WriteLine("Digite a opção desejada para buscar os Funcionários;");
        Console.WriteLine("1 = CLT; \n2= Aprendiz;");
        int busca = int.Parse(Console.ReadLine());
        ObterPorTipo(busca);
        opcaoValidado = true;
        ;break;
    case 2:
        Console.WriteLine("Digite ai o nome :D");
        string nomeMain = Console.ReadLine();
        ObterPorNome(nomeMain);
        opcaoValidado = true;
        ;break;
    case 3:
        ObterFuncionariosRecentes();
        opcaoValidado = true;
        ;break;
    case 4:
        ObterEstatisticas();
        opcaoValidado = true;
        ;break;
    case 5:
        bool fAdicionando = false;
        while(fAdicionando == false)
        {
            Funcionario f = new Funcionario();
            Console.WriteLine("Digite seu nome: ");
            f.Nome = Console.ReadLine();
        }
}
```

### 4) Opção 1: Escolha 1 CLT.



```
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
1
Digite a opção desejada para buscar os Funcionários;
1 = CLT;
2= Aprendiz;
1
Id: 1
Nome: Neymar
CPF: 12345678910
Admissão: 01/01/2000
Salário: R$ 100,00
Tipo: CLT
=====
Id: 2
Nome: Cristiano Ronaldo
CPF: 01987654321
Admissão: 30/06/2002
Salário: R$ 150,00
Tipo: CLT
=====
Id: 6
Nome: Roger Guedes
CPF: 246813579
Admissão: 13/12/1997
Salário: R$ 300,00
Tipo: CLT
=====
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
Program.cs 9+ x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ExibirCLT()
{
    1 reference
    public static void ObterPorTipo(int buscarFuncionario)
    {
        switch(buscarFuncionario)
        {
            case 1:ExibirCLT();break;
            case 2:ExibirAprendizes();break;
            default:
                Console.WriteLine($"O valor '{buscarFuncionario}' é invalido
                digite um valor valido");break;
        }
    }
    1 reference
    public static void ExibirAprendizes()
    {
        lista = lista.FindAll(x => x.TipoFuncionario ==
        TipoFuncionarioEnum.Aprendiz);
        ExibirLista();
    }
    1 reference
    public static void ExibirCLT()
    {
        lista = lista.FindAll(x => x.TipoFuncionario ==
        TipoFuncionarioEnum.CLT);
        ExibirLista();
    }
    8 references
    public static void ExibirLista()
    {
        string dados = "";
    }
}
```

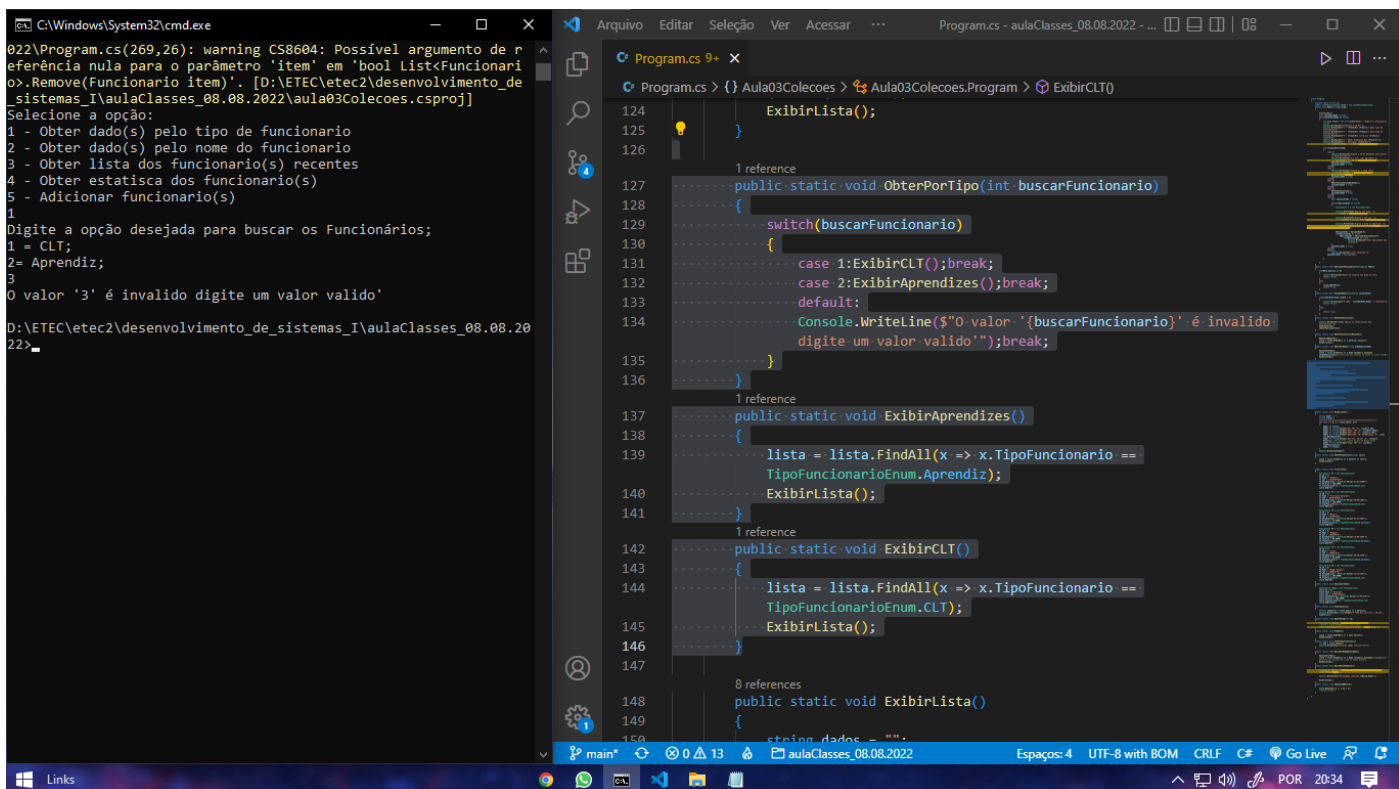
## 5) Opção 1: Escolha 2 Aprendiz.



```
C:\Windows\System32\cmd.exe
desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022\aula03Colecoes.csproj]
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
1
Digite a opção desejada para buscar os Funcionários;
1 = CLT;
2= Aprendiz;
2
=====
Id: 3
Nome: Messi
CPF: 135792468
Admissão: 01/11/2003
Salário: R$ 70,00
Tipo: Aprendiz
=====
Id: 4
Nome: Mbappe
CPF: 246813579
Admissão: 15/09/2005
Salário: R$ 80,00
Tipo: Aprendiz
=====
Id: 5
Nome: Lewa
CPF: 246813579
Admissão: 20/10/1998
Salário: R$ 90,00
Tipo: Aprendiz
=====
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>

Program.cs 9+ x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ExibirCLT()
124 ExibirLista();
125 }
126
127 1 reference
public static void ObterPorTipo(int buscarFuncionario)
128 {
129     switch(buscarFuncionario)
130     {
131         case 1:ExibirCLT();break;
132         case 2:ExibirAprendizes();break;
133         default:
134             Console.WriteLine($"O valor '{buscarFuncionario}' é invalido
digite um valor valido");break;
135     }
136 }
137 1 reference
public static void ExibirAprendizes()
138 {
139     lista = lista.FindAll(x => x.TipoFuncionario ==
TipoFuncionarioEnum.Aprendiz);
140     ExibirLista();
141 }
142 1 reference
public static void ExibirCLT()
143 {
144     lista = lista.FindAll(x => x.TipoFuncionario ==
TipoFuncionarioEnum.CLT);
145     ExibirLista();
146 }
147
148 8 references
public static void ExibirLista()
149 {
150     string dados = "";
```

## 6) Opção 1: com erro ao digitar o valor.



```
C:\Windows\System32\cmd.exe
022\Program.cs(269,26): warning CS8604: Possível argumento de r
eferência nula para o parâmetro 'item' em 'bool List<Funcionari
o>.Remove(Funcionario item)'. [D:\ETEC\etec2\desenvolvimento_de
_sistemas_I\aulaClasses_08.08.2022\aula03Colecoes.csproj]
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
1
Digite a opção desejada para buscar os Funcionários;
1 = CLT;
2= Aprendiz;
3
O valor '3' é invalido digite um valor valido'
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>

Program.cs 9+ x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ExibirCLT()
124 ExibirLista();
125 }
126
127 1 reference
public static void ObterPorTipo(int buscarFuncionario)
128 {
129     switch(buscarFuncionario)
130     {
131         case 1:ExibirCLT();break;
132         case 2:ExibirAprendizes();break;
133         default:
134             Console.WriteLine($"O valor '{buscarFuncionario}' é invalido
digite um valor valido");break;
135     }
136 }
137 1 reference
public static void ExibirAprendizes()
138 {
139     lista = lista.FindAll(x => x.TipoFuncionario ==
TipoFuncionarioEnum.Aprendiz);
140     ExibirLista();
141 }
142 1 reference
public static void ExibirCLT()
143 {
144     lista = lista.FindAll(x => x.TipoFuncionario ==
TipoFuncionarioEnum.CLT);
145     ExibirLista();
146 }
147
148 8 references
public static void ExibirLista()
149 {
150     string dados = "";
```

## 7) Opção 2: ObterPorNome() executado.

The screenshot shows a Windows command prompt window on the left and the Visual Studio IDE on the right. The command prompt displays the output of a .NET application, showing a menu of options and the details of a selected employee (Neymar). The Visual Studio window shows the source code of the application, with the `ObterPorNome()` method highlighted. The method is a public static void that takes a string parameter `nomeRequisitado` and returns a list of employees whose names contain the provided string. The code is written in C# and uses LINQ for filtering the list.

```
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
2
Digite ai o nome :D
neymar
=====
Id: 1
Nome: Neymar
CPF: 12345678910
Admissão: 01/01/2000
Salário: R$ 100,00
Tipo: CLT
=====
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>
```

```
Program.cs 9+ x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ObterPorNome(string nomeRequisitado)
95 > public static bool ValidarNome(Funcionario validarNome) ...
107 > public static void ObterEstatisticas() ...
113 >
114 > public static void ObterFuncionariosRecentes() ...
120 > public static void ObterPorNome(string nomeRequisitado)
121 > {
122 >     AdicionarItem();
123 >     lista = lista.FindAll(x => x.Nome.ToLower().Contains
124 >         (nomeRequisitado)); //Adiciona nos valores da lista os itens achados
125 >     ExibirLista();
126 > }
127 >
137 > public static void ExibirAprendizes() ...
142 > public static void ExibirCLT() ...
147 >
148 > public static void ExibirLista() ...
165 > public static void ObterPorSalario(decimal valor) ...
170 >
171 > public static void CriarLista() ...
227 > public static void AdicionarItem() ...
238 > public static void SomarSalario() ...
```

## 8) Opção 3: ObterFuncionariosRecentes() executado.

The screenshot shows a Windows command prompt window on the left and the Visual Studio IDE on the right. The command prompt displays the output of a .NET application, showing a menu of options and the details of two selected employees (Mbappe and Lewa). The Visual Studio window shows the source code of the application, with the `ObterFuncionariosRecentes()` method highlighted. The method is a public static void that returns a list of employees ordered by their salary in descending order. The code is written in C# and uses LINQ for ordering the list.

```
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
3
=====
Id: 4
Nome: Mbappe
CPF: 246813579
Admissão: 15/09/2005
Salário: R$ 80,00
Tipo: Aprendiz
=====
Id: 5
Nome: Lewa
CPF: 246813579
Admissão: 20/10/1998
Salário: R$ 90,00
Tipo: Aprendiz
=====
Id: 6
Nome: Roger Guedes
CPF: 246813579
Admissão: 13/12/1997
Salário: R$ 300,00
Tipo: CLT
=====
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.20
22>
```

```
Program.cs 9+ M x
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > RemoverIdMenor4()
82 > public static bool AdicionarFuncionario(Funcionario fNovo) ...
95 > public static bool ValidarNome(Funcionario validarNome) ...
107 > public static void ObterEstatisticas() ...
113 >
114 > public static void ObterFuncionariosRecentes()
115 > {
116 >     RemoverIdMenor4();
117 >     lista = lista.OrderBy(x => x.Salario).ToList();
118 >     ExibirLista();
119 > }
120 > public static void RemoverIdMenor4()
121 > {
122 >     lista.RemoveAll(x => x.Id < 4);
123 >     //ExibirLista();
124 > }
125 > public static void ObterPorNome(string nomeRequisitado) ...
131 >
132 > public static void ObterPorTipo(int buscarFuncionario) ...
142 > public static void ExibirAprendizes() ...
147 > public static void ExibirCLT() ...
152 >
153 > public static void ExibirLista() ...
170 > public static void ObterPorSalario(decimal valor) ...
```

## 9) Opção 4: ObterEstatisticas() executado.

The screenshot shows a Windows command prompt window on the left and the Visual Studio IDE on the right. The command prompt displays the output of the program, showing a warning about a null reference, a list of menu options, and the results of the 'ObterEstatisticas()' method: 'A soma dos salários é R\$ 790,00' and 'Existem 6 funcionários'. The Visual Studio window shows the source code of the 'Program.cs' file, with the 'ObterEstatisticas()' method highlighted. The method contains three lines of code: 'Console.WriteLine("Segue abaixo as estatísticas dos Funcionários:");', 'SomarSalario();', and 'ContarFuncionarios();'.

```
022\Program.cs(273,26): warning CS8604: Possível argumento de referência nula para o parâmetro 'item' em 'bool list<Funcionario>.Remove(Funcionario item)'. [D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022\aula03Colecoes.csproj]
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
4
Segue abaixo as estatísticas dos Funcionários:
A soma dos salários é R$ 790,00
Existem 6 funcionários

D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>
```

```
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > RemoverIdMenor4()
82 > public static bool AdicionarFuncionario(Funcionario fNovo) ...
95 > public static bool ValidarNome(Funcionario validarNome) ...
107 > public static void ObterEstatisticas()
108 {
109     Console.WriteLine("Segue abaixo as estatísticas dos
110     Funcionários:");
111     SomarSalario();
112     ContarFuncionarios();
113 }
114 > public static void ObterFuncionariosRecentes() ...
120 > public static void RemoverIdMenor4() ...
125 > public static void ObterPorNome(string nomeRequisitado) ...
131 >
132 > public static void ObterPorTipo(int buscarFuncionario) ...
142 > public static void ExibirAprendizes() ...
147 > public static void ExibirCLT() ...
152 >
153 > public static void ExibirLista() ...
170 > public static void ObterPorSalario(decimal valor) ...
175 >
176 > public static void CriarLista() ...
```

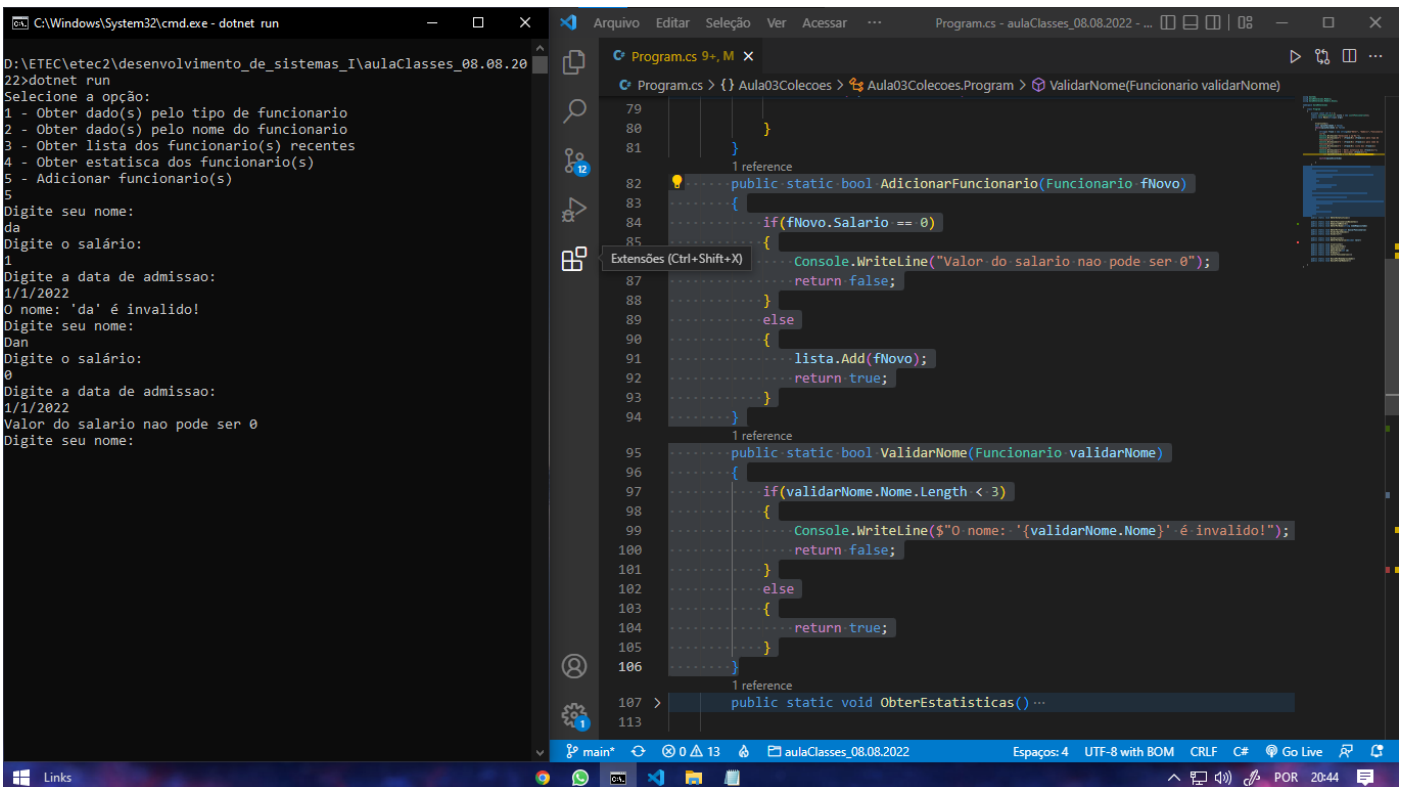
## 10) Opção 5: ValidarNome() apresentando erro.

The screenshot shows a Windows command prompt window on the left and the Visual Studio IDE on the right. The command prompt displays the output of the program, showing a list of menu options, the user input 'da', and the error message 'O nome: 'da' é inválido!'. The Visual Studio window shows the source code of the 'Program.cs' file, with the 'ValidarNome()' method highlighted. The method contains two lines of code: 'Console.WriteLine(\$"O nome: '{validarNome.Nome}' é inválido!");' and 'return false;'.

```
D:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatística dos funcionario(s)
5 - Adicionar funcionario(s)
5
Digite seu nome:
da
Digite o salário:
1
Digite a data de admissao:
1/1/2022
O nome: 'da' é inválido!
Digite seu nome:
-
```

```
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ValidarNome(Funcionario validarNome)
80 > }
81 > }
82 > public static bool AdicionarFuncionario(Funcionario fNovo)
83 {
84     if(fNovo.Salario == 0)
85     {
86         Console.WriteLine("Valor do salario nao pode ser 0");
87         return false;
88     }
89     else
90     {
91         lista.Add(fNovo);
92         return true;
93     }
94 }
95 > public static bool ValidarNome(Funcionario validarNome)
96 {
97     if(validarNome.Nome.Length < 3)
98     {
99         Console.WriteLine($"O nome: '{validarNome.Nome}' é inválido!");
100         return false;
101     }
102     else
103     {
104         return true;
105     }
106 }
107 > public static void ObterEstatisticas() ...
113 >
```

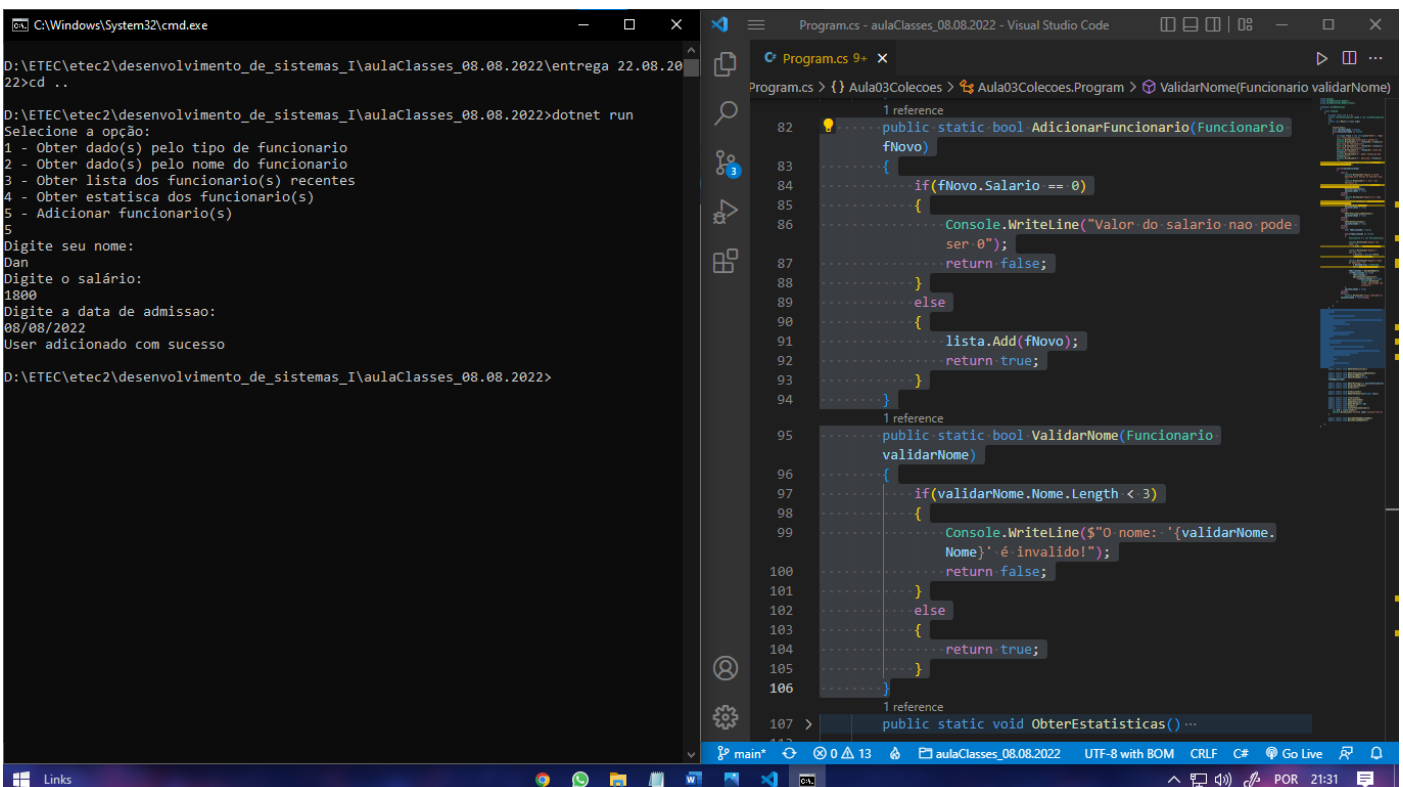
## 11) Opção 5: Erro ao digitar salário.



```
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>dotnet run
22>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatistica dos funcionario(s)
5 - Adicionar funcionario(s)
5
Digite seu nome:
da
Digite o salário:
1
Digite a data de admissao:
1/1/2022
0 nome: 'da' é invalido!
Digite seu nome:
Dan
Digite o salário:
0
Digite a data de admissao:
1/1/2022
Valor do salario nao pode ser 0
Digite seu nome:
```

```
Program.cs - aulaClasses_08.08.2022 - ...
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ValidarNome(Funcionario validarNome)
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
113
```

## 12) Opção 5: Sem apresentar erros.



```
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>entrega 22.08.2022>cd ..
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>dotnet run
Selecione a opção:
1 - Obter dado(s) pelo tipo de funcionario
2 - Obter dado(s) pelo nome do funcionario
3 - Obter lista dos funcionario(s) recentes
4 - Obter estatistica dos funcionario(s)
5 - Adicionar funcionario(s)
5
Digite seu nome:
Dan
Digite o salário:
1800
Digite a data de admissao:
08/08/2022
User adicionado com sucesso
0:\ETEC\etec2\desenvolvimento_de_sistemas_I\aulaClasses_08.08.2022>
```

```
Program.cs - aulaClasses_08.08.2022 - Visual Studio Code
Program.cs > {} Aula03Colecoes > Aula03Colecoes.Program > ValidarNome(Funcionario validarNome)
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
```