

Missão Prática 4

Execução do Projeto

Configurações da Máquina

A atividade foi executada através de um desktop com o sistema operacional Ubuntu 22.04 LTS.

1. Microatividade 1

The image shows the Visual Studio Code editor interface. The main editor window displays a Python file named `Pessoa.py` with the following code:

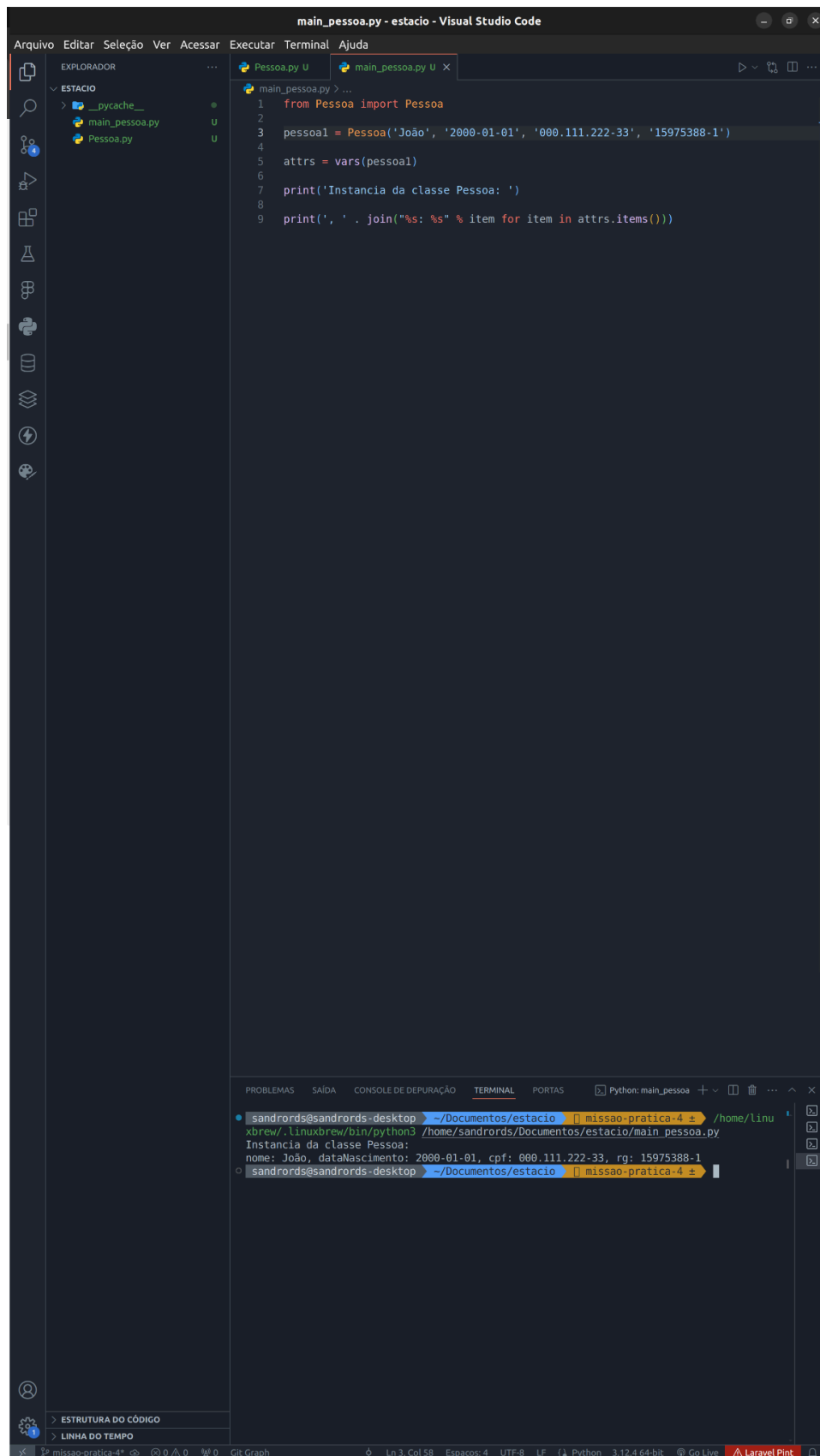
```
1 class Pessoa:
2     def __init__(self, nome, dataNascimento, cpf, rg):
3         self.nome = nome
4         self.dataNascimento = dataNascimento
5         self.cpf = cpf
6         self.rg = rg;
```

The sidebar on the left contains the Explorer, Search, and Run and Debug views. The bottom panel shows the Terminal with a command prompt and a file explorer view showing the project structure:

```
/home/linuxbrew/.linuxbrew/bin/python3 /home/sandrorods/Documentos/estacio/Pessoa.py
sandrorods@sandrorods-desktop ~ - /Documentos/estacio - 11:53:04 AM - /home/linu
linuxbrew/.linuxbrew/bin/python3 /home/sandrorods/Documentos/estacio/Pessoa.py
sandrorods@sandrorods-desktop ~ - /Documentos/estacio - 11:53:04 AM - /home/linu
```

Criando a classe Pessoa.

2. Microatividade 2



The screenshot displays the Visual Studio Code interface with the file explorer on the left showing a project named 'ESTACIO' containing files like '__pycache__', 'main_pessoa.py', and 'Pessoa.py'. The main editor window shows the code for 'main_pessoa.py' with the following Python code:

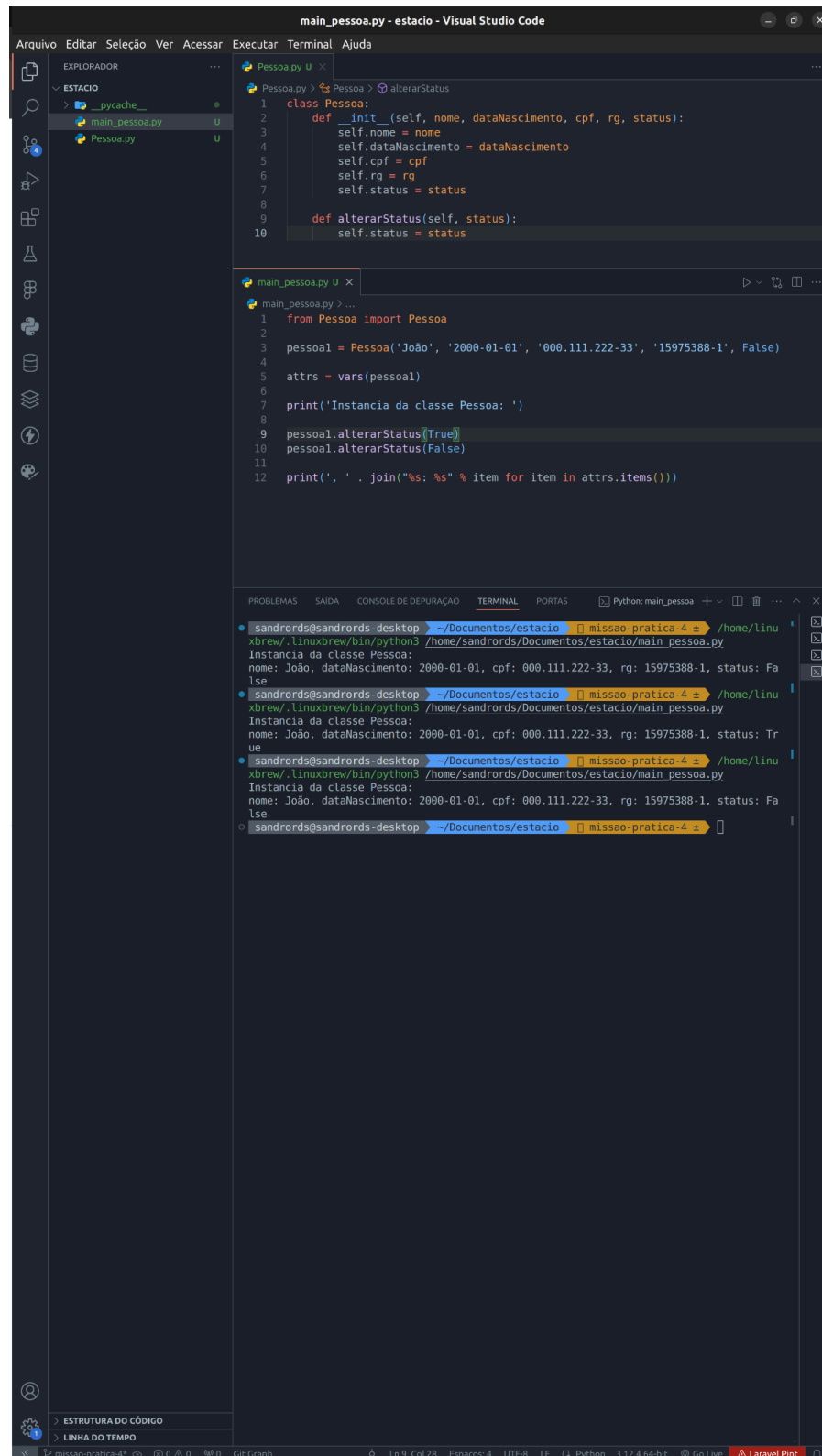
```
1 from Pessoa import Pessoa
2
3 pessoa1 = Pessoa('João', '2000-01-01', '000.111.222-33', '15975388-1')
4
5 attrs = vars(pessoa1)
6
7 print('Instancia da classe Pessoa: ')
8
9 print(', '.join("%s: %s" % item for item in attrs.items()))
```

The bottom panel shows the terminal output, indicating the script was executed successfully from the directory ~/Documentos/estacio. The output of the script is:

```
Instancia da classe Pessoa:
nome: João, dataNascimento: 2000-01-01, cpf: 000.111.222-33, rg: 15975388-1
```

Criando o script main_pessoa.

3. Microatividade 3



The screenshot displays the Visual Studio Code interface with two Python files open. The top file, `Pessoa.py`, defines a `Pessoa` class with an `__init__` method that initializes `nome`, `dataNascimento`, `cpf`, `rg`, and `status` attributes. It also includes an `alterarStatus` method to update the `status` attribute. The bottom file, `main_pessoa.py`, imports the `Pessoa` class, creates an instance named `peessoa` with specific attributes, and prints the instance details. It then demonstrates the `alterarStatus` method by setting the status to `True` and `False`, and finally prints the updated attributes.

```
main_pessoa.py - estacio - Visual Studio Code

Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda

EXPLORADOR
ESTACIO
  _pycache_
  main_pessoa.py
  Pessoa.py

Pessoa.py
1 class Pessoa:
2     def __init__(self, nome, dataNascimento, cpf, rg, status):
3         self.nome = nome
4         self.dataNascimento = dataNascimento
5         self.cpf = cpf
6         self.rg = rg
7         self.status = status
8
9     def alterarStatus(self, status):
10        self.status = status

main_pessoa.py
1 from Pessoa import Pessoa
2
3 peessoa = Pessoa('João', '2000-01-01', '000.111.222-33', '15975388-1', False)
4
5 attrs = vars(peessoa)
6
7 print('Instancia da classe Pessoa: ')
8
9 peessoa.alterarStatus(True)
10 peessoa.alterarStatus(False)
11
12 print(', '.join("%s: %s" % item for item in attrs.items()))

PROBLEMAS  SAÍDA  CONSOLE DE DEBURAÇÃO  TERMINAL  PORTAS  Python: main_pessoa
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/linu
xbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
Instancia da classe Pessoa:
nome: João, dataNascimento: 2000-01-01, cpf: 000.111.222-33, rg: 15975388-1, status: Fa
lse
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/linu
xbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
Instancia da classe Pessoa:
nome: João, dataNascimento: 2000-01-01, cpf: 000.111.222-33, rg: 15975388-1, status: Tr
ue
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/linu
xbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
Instancia da classe Pessoa:
nome: João, dataNascimento: 2000-01-01, cpf: 000.111.222-33, rg: 15975388-1, status: Fa
lse
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4]

> ESTRUTURA DO CÓDIGO
> LINHA DO TEMPO
```

Adicionando a propriedade status e o método alterarStatus.

4. Microatividade 4

```
Pessoa.py - estacio - Visual Studio Code

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

EXPLORADOR
ESTACIO
  __pycache__
  main_pessoa.py
  Pessoa.py

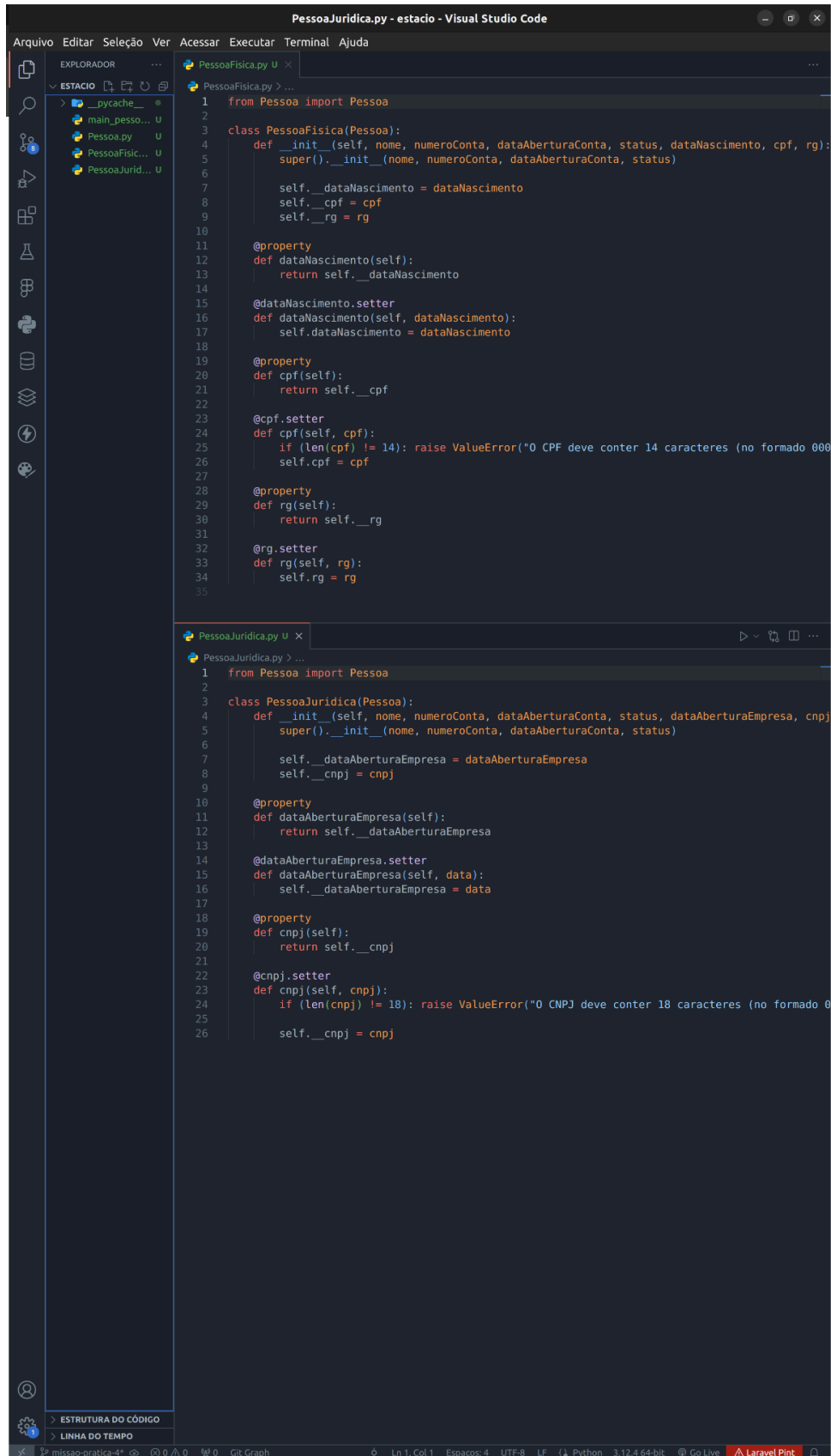
main_pessoa.py
1 from Pessoa import Pessoa
2
3 pessoa1 = Pessoa('João', '2000-01-01', '000.111.222-33', '15975388-1', False)
4
5 print(' '.join("%s: %s" % item for item in vars(pessoa1).items()))
6
7 pessoa1.nome = 'Ana'
8 pessoa1.dataNascimento = '2002-03-20'
9 pessoa1.cpf = '111.222.333-44'
10 pessoa1.rg = '32165412-9'
11 pessoa1.status = True
12
13 print(' '.join("%s: %s" % item for item in vars(pessoa1).items()))
14

Pessoa.py
1 class Pessoa:
2     def __init__(self, nome, dataNascimento, cpf, rg, status):
3         self.__nome = nome
4         self.__dataNascimento = dataNascimento
5         self.__cpf = cpf
6         self.__rg = rg
7         self.__status = status
8
9     @property
10    def nome(self):
11        return self.__nome
12
13    @nome.setter
14    def nome(self, nome):
15        self.__nome = nome
16
17    @property
18    def dataNascimento(self):
19        return self.__dataNascimento
20
21    @dataNascimento.setter
22    def dataNascimento(self, dataNascimento):
23        self.__dataNascimento = dataNascimento
24
25    @property
26    def cpf(self):
27        return self.__cpf
28
29    @cpf.setter
30    def cpf(self, cpf):
31        if (len(cpf) != 14): raise ValueError("0 CPF deve conter 14 caracteres (no fo
32        self.__cpf = cpf
33
34    @property
35    def rg(self):
36        return self.__rg
37
38    @rg.setter
39    def rg(self, rg):
40        self.__rg = rg
41
42    @property
43    def status(self):
44        return self.__status
45
46    @status.setter
47    def status(self, status):
48        self.__status = status

TERMINAL
Python: main_pessoa
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/linu
xbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
_Pessoa_nome: João, _Pessoa_dataNascimento: 2000-01-01, _Pessoa_cpf: 000.111.222-33,
_Pessoa_rg: 15975388-1, _Pessoa_status: False
Traceback (most recent call last):
  File "/home/sandrords/Documentos/estacio/main_pessoa.py", line 9, in <module>
    pessoa1.cpf = '00011122233'
    ~~~~~
File "/home/sandrords/Documentos/estacio/Pessoa.py", line 31, in cpf
    if (len(cpf) != 14): raise ValueError("0 CPF deve conter 14 caracteres (no formato
000.000.000-00)")
ValueError: 0 CPF deve conter 14 caracteres (no formato 000.000.000-00).
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/li
nuxbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
_Pessoa_nome: João, _Pessoa_dataNascimento: 2000-01-01, _Pessoa_cpf: 000.111.222-33,
_Pessoa_rg: 15975388-1, _Pessoa_status: False
_Pessoa_nome: Ana, _Pessoa_dataNascimento: 2002-03-20, _Pessoa_cpf: 111.222.333-44,
_Pessoa_rg: 32165412-9, _Pessoa_status: True
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4]
```

Criando os getters e setters da classe Pessoa.

5. Microatividade 5



The image shows a Visual Studio Code editor window titled "PessoaJuridica.py - estacio - Visual Studio Code". The editor displays two Python files: `PessoaFisica.py` and `PessoaJuridica.py`.

PessoaFisica.py (lines 1-35):

```
1 from Pessoa import Pessoa
2
3 class PessoaFisica(Pessoa):
4     def __init__(self, nome, numeroConta, dataAberturaConta, status, dataNascimento, cpf, rg):
5         super().__init__(nome, numeroConta, dataAberturaConta, status)
6
7         self.__dataNascimento = dataNascimento
8         self.__cpf = cpf
9         self.__rg = rg
10
11 @property
12 def dataNascimento(self):
13     return self.__dataNascimento
14
15 @dataNascimento.setter
16 def dataNascimento(self, dataNascimento):
17     self.__dataNascimento = dataNascimento
18
19 @property
20 def cpf(self):
21     return self.__cpf
22
23 @cpf.setter
24 def cpf(self, cpf):
25     if (len(cpf) != 14): raise ValueError("0 CPF deve conter 14 caracteres (no formado 000
26     self.__cpf = cpf
27
28 @property
29 def rg(self):
30     return self.__rg
31
32 @rg.setter
33 def rg(self, rg):
34     self.__rg = rg
35
```

PessoaJuridica.py (lines 1-26):

```
1 from Pessoa import Pessoa
2
3 class PessoaJuridica(Pessoa):
4     def __init__(self, nome, numeroConta, dataAberturaConta, status, dataAberturaEmpresa, cnpj)
5         super().__init__(nome, numeroConta, dataAberturaConta, status)
6
7         self.__dataAberturaEmpresa = dataAberturaEmpresa
8         self.__cnpj = cnpj
9
10 @property
11 def dataAberturaEmpresa(self):
12     return self.__dataAberturaEmpresa
13
14 @dataAberturaEmpresa.setter
15 def dataAberturaEmpresa(self, data):
16     self.__dataAberturaEmpresa = data
17
18 @property
19 def cnpj(self):
20     return self.__cnpj
21
22 @cnpj.setter
23 def cnpj(self, cnpj):
24     if (len(cnpj) != 18): raise ValueError("0 CNPJ deve conter 18 caracteres (no formado 0
25
26     self.__cnpj = cnpj
27
```

The status bar at the bottom indicates the file is "missao-pratica-4*", encoding is "UTF-8", and the language is "Python".

Criação das Classes PessoaFisica e PessoaJuridica.

The image shows a Visual Studio Code editor window titled "main_pessoa.py - estacio - Visual Studio Code". The interface includes a sidebar on the left with icons for Explorer, Search, Run and Debug, and Extensions. The Explorer view shows a file structure with "ESTACIO" as a folder and several Python files: "Pessoa.py", "main_pessoa.py", "PessoaFisica.py", and "PessoaJuridica.py".

The main editor area displays two files. The top file, "main_pessoa.py", contains the following Python code:

```
1 from Pessoa import Pessoa
2 from PessoaFisica import PessoaFisica
3 from PessoaJuridica import PessoaJuridica
4
5 pessoa = Pessoa('João', '1239819238', '2003-05-21', False)
6 pessoa.mostrarDados()
7
8 pessoaFisica = PessoaFisica('João', '1239819238', '2003-05-21', False, '1980-03-21', '000.111.111-00')
9 pessoaFisica.mostrarDados()
10
11 pessoaJuridica = PessoaJuridica('João', '1239819238', '2003-05-21', False, '2010-03-01', '32.356/0001-09')
12 pessoaJuridica.mostrarDados()
```

The bottom file, "Pessoa.py", defines the "Pessoa" class with the following code:

```
1 class Pessoa:
2     def __init__(self, nome, numeroConta, dataAberturaConta, status):
3         self.__nome = nome
4         self.__numeroConta = numeroConta
5         self.__dataAberturaConta = dataAberturaConta
6         self.__status = status
7
8     def mostrarDados(self):
9         print()
10        print(' '.join("%s: %s" % item for item in vars(self).items()))
11
12    @property
13    def nome(self):
14        return self.__nome
15
16    @nome.setter
17    def nome(self, nome):
18        self.__nome = nome
19
20    @property
21    def numeroConta(self):
22        return self.__numeroConta
23
24    @numeroConta.setter
25    def numeroConta(self, numeroConta):
26        self.__numeroConta = numeroConta
27
28    @property
29    def dataAberturaConta(self):
30        return self.__dataAberturaConta
31
32    @dataAberturaConta.setter
33    def dataAberturaConta(self, dataAberturaConta):
34        self.__dataAberturaConta = dataAberturaConta
35
36    @property
37    def status(self):
38        return self.__status
39
40    @status.setter
41    def status(self, status):
42        self.__status = status
```

The bottom panel of the editor shows the "TERMINAL" view with the following output:

```
Python: main_pessoa
[sandrords@sandrords-desktop ~/Documentos/estacio] missao-pratica-4 ± /home/linuxbrew/.lin
uxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _P
essoa_status: False
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _P
essoa_status: False, _PessoaFisica_dataNascimento: 1980-03-21, _PessoaFisica_cpf: 000.111.222-
33, _PessoaFisica_rg: 123654789-0
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _P
essoa_status: False, _PessoaJuridica_dataAberturaEmpresa: 2010-03-01, _PessoaJuridica_cnpj: 32
.324.356/0001-09
[sandrords@sandrords-desktop ~/Documentos/estacio] missao-pratica-4 ±
```

Criação da classe Pessoa e execução do script main_pessoa.

The image shows a Visual Studio Code editor window titled "main_pessoa.py - estacio - Visual Studio Code". The editor is open to a file named "main_pessoa.py" in the "ESTACIO" workspace. The code defines three classes: `Pessoa`, `PessoaFisica`, and `PessoaJuridica`. The `Pessoa` class has attributes `nome`, `numeroConta`, `dataAberturaConta`, and `status`. The `PessoaFisica` class inherits from `Pessoa` and adds `dataNascimento`, `cpf`, and `rg`. The `PessoaJuridica` class inherits from `Pessoa` and adds `dataAberturaEmpresa` and `cnpj`. The script creates instances of these classes and calls the `mostrarDados()` method for each.

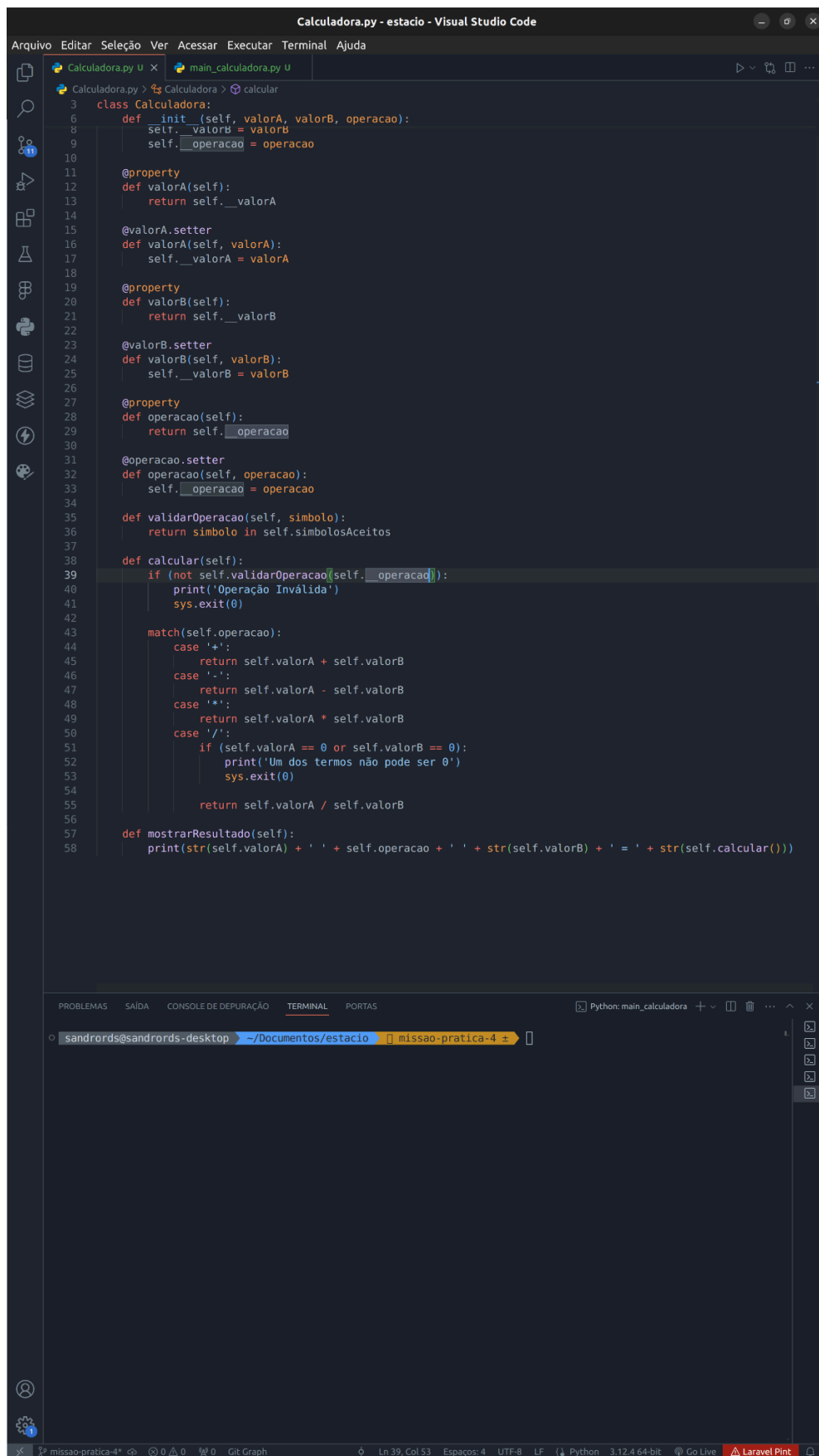
```
1 from Pessoa import Pessoa
2 from PessoaFisica import PessoaFisica
3 from PessoaJuridica import PessoaJuridica
4
5 pessoa = Pessoa('João', '1239819238', '2003-05-21', False)
6 pessoa.mostrarDados()
7
8 pessoaFisica = PessoaFisica('João', '1239819238', '2003-05-21', False, '1980-03-21', '000.111.222-33')
9 pessoaFisica.mostrarDados()
10
11 pessoaJuridica = PessoaJuridica('João', '1239819238', '2003-05-21', False, '2010-03-01', '32.356/0001-09')
12 pessoaJuridica.mostrarDados()
13
14 pessoa.nome = 'Maria'
15 pessoa.numeroConta = '8192381239'
16 pessoa.dataAberturaConta = '2005-06-11'
17 pessoa.status = True
18 pessoa.mostrarDados()
19
20 pessoaFisica.nome = 'Maria'
21 pessoaFisica.numeroConta = '8192381239'
22 pessoaFisica.dataAberturaConta = '2005-06-11'
23 pessoaFisica.status = True
24 pessoaFisica.dataNascimento = '2002-06-07'
25 pessoaFisica.cpf = '819.937.483-23'
26 pessoaFisica.rg = '18274637-9'
27 pessoaFisica.mostrarDados()
28
29 pessoaJuridica.nome = 'Maria'
30 pessoaJuridica.numeroConta = '8192381239'
31 pessoaJuridica.dataAberturaConta = '2005-06-11'
32 pessoaJuridica.status = True
33 pessoaJuridica.dataAberturaEmpresa = '2002-06-07'
34 pessoaJuridica.cnpj = '12.335.243/2001-00'
35 pessoaJuridica.mostrarDados()
36
37
```

The terminal output shows the results of the `mostrarDados()` method calls for each instance, displaying the attributes and their values.

```
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] /home/linuxbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_pessoa.py
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _Pessoa_status: False
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _Pessoa_status: False, _PessoaFisica_dataNascimento: 1980-03-21, _PessoaFisica_cpf: 000.111.222-33, _PessoaFisica_rg: 123654789-0
_Pessoa_nome: João, _Pessoa_numeroConta: 1239819238, _Pessoa_dataAberturaConta: 2003-05-21, _Pessoa_status: False, _PessoaJuridica_dataAberturaEmpresa: 2010-03-01, _PessoaJuridica_cnpj: 32.356/0001-09
_Pessoa_nome: Maria, _Pessoa_numeroConta: 8192381239, _Pessoa_dataAberturaConta: 2005-06-11, _Pessoa_status: True
_Pessoa_nome: Maria, _Pessoa_numeroConta: 8192381239, _Pessoa_dataAberturaConta: 2005-06-11, _Pessoa_status: True
_Pessoa_nome: Maria, _Pessoa_numeroConta: 8192381239, _Pessoa_dataAberturaConta: 2005-06-11, _Pessoa_status: True, _PessoaFisica_dataNascimento: 2002-06-07, _PessoaFisica_cpf: 819.937.483-23, _PessoaFisica_rg: 18274637-9
_Pessoa_nome: Maria, _Pessoa_numeroConta: 8192381239, _Pessoa_dataAberturaConta: 2005-06-11, _Pessoa_status: True, _PessoaJuridica_dataAberturaEmpresa: 2002-06-07, _PessoaJuridica_cnpj: 12.335.243/2001-00
```

Trocando os valores de cada objeto.

Missão Prática



The image shows a Visual Studio Code editor window titled "Calculadora.py - estacio - Visual Studio Code". The editor displays a Python script for a calculator class. The code includes an initialization method, property getters and setters for two values and an operation, a validation method, a calculation method using a match statement, and a method to display the result. The terminal at the bottom shows the command prompt and the file path.

```
3 class Calculadora:
4     def __init__(self, valorA, valorB, operacao):
5         self.__valorA = valorA
6         self.__valorB = valorB
7         self.__operacao = operacao
8
9
10
11 @property
12 def valorA(self):
13     return self.__valorA
14
15 @valorA.setter
16 def valorA(self, valorA):
17     self.__valorA = valorA
18
19 @property
20 def valorB(self):
21     return self.__valorB
22
23 @valorB.setter
24 def valorB(self, valorB):
25     self.__valorB = valorB
26
27 @property
28 def operacao(self):
29     return self.__operacao
30
31 @operacao.setter
32 def operacao(self, operacao):
33     self.__operacao = operacao
34
35 def validarOperacao(self, simbolo):
36     return simbolo in self.simbolosAceitos
37
38 def calcular(self):
39     if (not self.validarOperacao(self.__operacao)):
40         print('Operação Inválida')
41         sys.exit(0)
42
43     match(self.operacao):
44         case '+':
45             return self.valorA + self.valorB
46         case '-':
47             return self.valorA - self.valorB
48         case '*':
49             return self.valorA * self.valorB
50         case '/':
51             if (self.valorA == 0 or self.valorB == 0):
52                 print('Um dos termos não pode ser 0')
53                 sys.exit(0)
54             return self.valorA / self.valorB
55
56 def mostrarResultado(self):
57     print(str(self.valorA) + ' ' + self.operacao + ' ' + str(self.valorB) + ' = ' + str(self.calcular()))
58
```

Terminal output:

```
sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-4
```

Implementação da Classe Calculadora

The image shows a Visual Studio Code window titled "main_calculadora.py - estacio - Visual Studio Code". The Explorer sidebar on the left shows a project named "ESTACIO" with files: "__pycache__", "Calculadora...", "main_calcul...", "main_pesso...", "Pessoa.py", "PessoaFisic...", and "PessoaJurid...". The main editor displays the file "main_calculadora.py" with the following Python code:

```
1 from Calculadora import Calculadora
2
3 Calculadora(6, 2, '+').mostrarResultado()
4 Calculadora(6, 2, '-').mostrarResultado()
5 Calculadora(6, 2, '*').mostrarResultado()
6 Calculadora(6, 2, '/').mostrarResultado()
7 Calculadora(6, 0, '/').mostrarResultado()
```

The TERMINAL panel at the bottom shows the command prompt output:

```
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] $ /home/linuxbrew/.lin
uxbrew/bin/python3 /home/sandrords/Documentos/estacio/main_calculadora.py
6 + 2 = 8
6 - 2 = 4
6 * 2 = 12
6 / 2 = 3.0
Um dos termos não pode ser 0
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4] $
```

The status bar at the bottom indicates the file is "missao-pratica-4*", the cursor is at "Ln 7, Col 42", the encoding is "UTF-8", and the language is "Python 3.12.4 64-bit".

Implementação da primeira versão, sem prompt

Calculadora.py - estacio - Visual Studio Code

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

Calculadora.py U x main_calculadora.py U

```
1 import sys
2
3 class Calculadora:
4     simbolosAceitos = ['+', '-', '*', '/']
5
6     def __init__(self):
7         self.__valorA = float(input('Primeiro termo: '))
8         self.__valorB = float(input('Segundo termo: '))
9         self.__operacao = input('Operação: ')
10
11     @property
12     def valorA(self):
13         return self.__valorA
14
15     @valorA.setter
16     def valorA(self, valorA):
17         self.__valorA = valorA
18
19     @property
20     def valorB(self):
21         return self.__valorB
22
23     @valorB.setter
24     def valorB(self, valorB):
25         self.__valorB = valorB
26
27     @property
28     def operacao(self):
29         return self.__operacao
30
31     @operacao.setter
32     def operacao(self, operacao):
33         self.__operacao = operacao
34
35     def validarOperacao(self, simbolo):
36         return simbolo in self.simbolosAceitos
37
38     def calcular(self):
39         if (not self.validarOperacao(self.__operacao)):
40             print('Operação Inválida')
41             sys.exit(0)
42
43         match(self.operacao):
44             case '+':
45                 return self.valorA + self.valorB
46             case '-':
47                 return self.valorA - self.valorB
48             case '*':
49                 return self.valorA * self.valorB
50             case '/':
51                 if (self.valorA == 0 or self.valorB == 0):
52                     print('Um dos termos não pode ser 0')
53                     sys.exit(0)
54                 return self.valorA / self.valorB
55
56     def mostrarResultado(self):
57         print(str(self.valorA) + ' ' + self.operacao + ' ' + str(self.valorB) + ' = ' + str(self.calcular()))
58
```

PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO TERMINAL PORTAS

Python: main_calculadora

```
● sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-4 + /home/linuxbrew/.linuxbrew/bin/python3 /h
ome/sandrords/Documentos/estacio/main_calculadora.py
Primeiro termo: 1
Segundo termo: 2
Operação: +
1.0 + 2.0 = 3.0
Primeiro termo: 6
Segundo termo: 2
Operação: -
6.0 - 2.0 = 4.0
Primeiro termo: 6
Segundo termo: 2
Operação: *
6.0 * 2.0 = 12.0
Primeiro termo: 6
Segundo termo: 2
Operação: /
6.0 / 2.0 = 3.0
Primeiro termo: 6
Segundo termo: 0
Operação: /
Um dos termos não pode ser 0
○ sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-4 +
```

missao-pratica-4 Ln 9, Col 46 Espaços: 4 UTF-8 LF Python 3.12.4 64-bit Go Live Laravel Pint

Implementação da Classe Calculadora com funcionalidade de obter dados via Prompt

The image shows a Visual Studio Code window titled "main_calculadora.py - estacio - Visual Studio Code". The editor displays a Python file named "main_calculadora.py" with the following code:

```
1 from Calculadora import Calculadora
2
3 Calculadora().mostrarResultado()
4 Calculadora().mostrarResultado()
5 Calculadora().mostrarResultado()
6 Calculadora().mostrarResultado()
7 Calculadora().mostrarResultado()
```

The bottom panel shows the "TERMINAL" output, which displays the execution of the script. The output shows the results of the five calls to the `mostrarResultado()` method, including prompts for terms and operations, and the resulting calculations.

```
• sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4 ±] /home/linuxbrew/.linuxbrew/bin/python3 /h
ome/sandrords/Documentos/estacio/main_calculadora.py
Primeiro termo: 1
Segundo termo: 2
Operação: +
1.0 + 2.0 = 3.0
Primeiro termo: 6
Segundo termo: 2
Operação: -
6.0 - 2.0 = 4.0
Primeiro termo: 6
Segundo termo: 2
Operação: *
6.0 * 2.0 = 12.0
Primeiro termo: 6
Segundo termo: 2
Operação: /
6.0 / 2.0 = 3.0
Primeiro termo: 6
Segundo termo: 0
Operação: /
Um dos termos não pode ser 0
○ sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-4 ±]
```

Chamadas da Classe Calculadora no script principal