

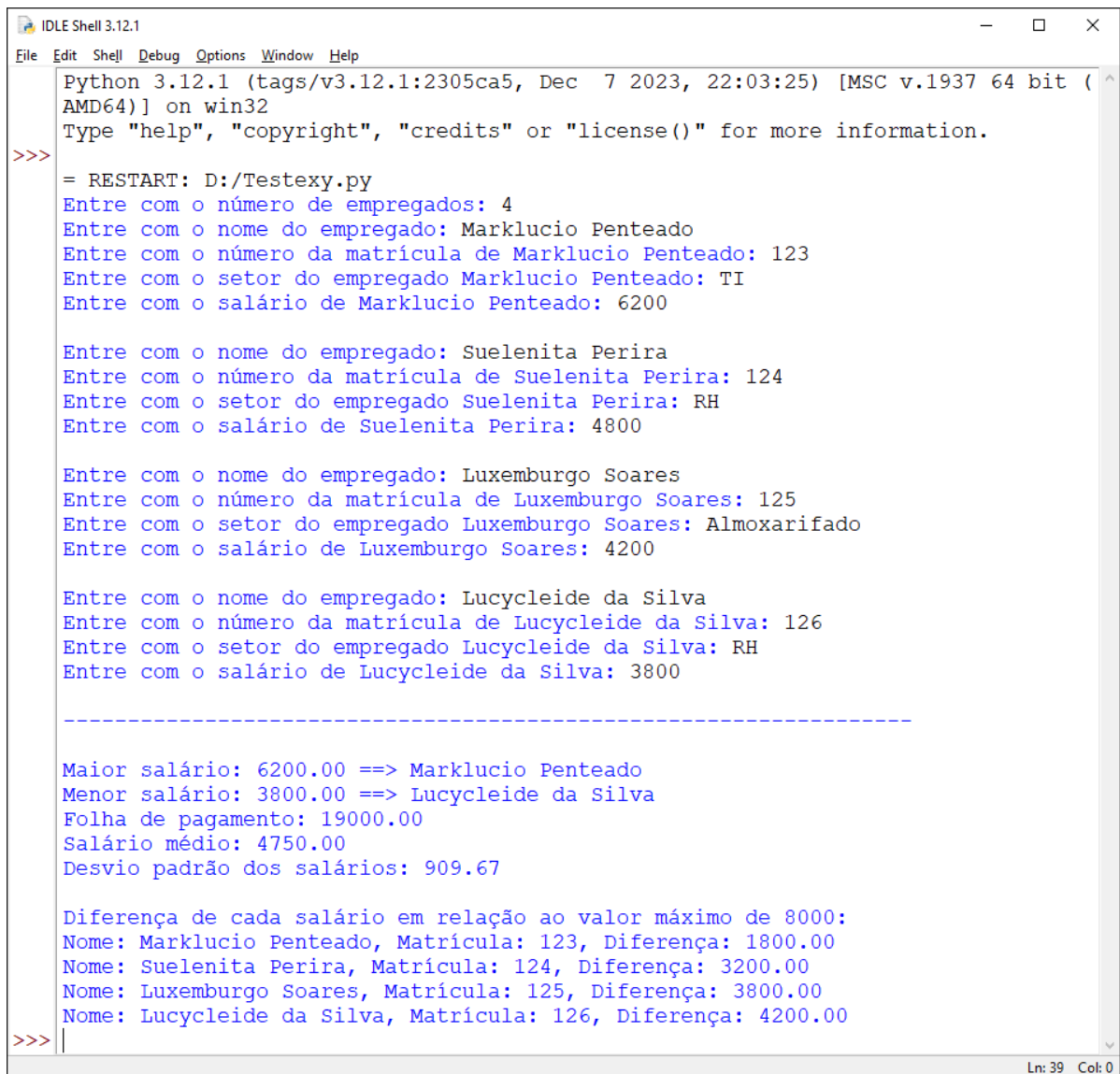
# Controle Salarial

## Mário Leite

...

A Folha de Pagamento pode ser uma tremenda dor de cabeça para os pequenos empresários tipo MEI, na hora de encarar a realidade financeira de suas empresas, no Brasil. Além de ter que arcar com as despesas normais com suprimentos e matérias primas, o empresário tem que considerar que um único funcionário de sua empresa custa quase o dobro do salário dele, em forma de obrigações trabalhistas. Os sistemas que implementam programas computacionais que gerenciam e controlam a Folha de Pagamento das empresas são muito importantes, chegando até, serem vendidos como *softwares* de “prateleira”; expressão muito comum na década dos anos 80 do século passado, vendidos com implementações em Clipper: o ambiente/linguagem mais popular na época. Hoje em dia esse tipo de sistema pode ser encontrado em diversas linguagens de programação, sobretudo em Python: uma das mais populares atualmente.

O programa **“ControleSalarial”**, embora bem simples e com apenas quatro empregados (para não poluir muito a saída) poderia, como exemplo, ser uma boa implementação para os programadores iniciantes, pois mostra como implementar uma solução básica para o dono da empresa ter uma ideia como anda a Folha de Pagamento com relação aos salários de seus empregados. É uma solução bem simples, mas que pode guiar o programador para criar soluções mais sofisticadas como por exemplo, o quão os salários estão afastados de um salário médio base que a empresa pode pagar.



```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec 7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Testexy.py
Entre com o número de empregados: 4
Entre com o nome do empregado: Marklucio Penteado
Entre com o número da matrícula de Marklucio Penteado: 123
Entre com o setor do empregado Marklucio Penteado: TI
Entre com o salário de Marklucio Penteado: 6200

Entre com o nome do empregado: Suelenita Perira
Entre com o número da matrícula de Suelenita Perira: 124
Entre com o setor do empregado Suelenita Perira: RH
Entre com o salário de Suelenita Perira: 4800

Entre com o nome do empregado: Luxemburgo Soares
Entre com o número da matrícula de Luxemburgo Soares: 125
Entre com o setor do empregado Luxemburgo Soares: Almoxarifado
Entre com o salário de Luxemburgo Soares: 4200

Entre com o nome do empregado: Lucycleide da Silva
Entre com o número da matrícula de Lucycleide da Silva: 126
Entre com o setor do empregado Lucycleide da Silva: RH
Entre com o salário de Lucycleide da Silva: 3800

-----

Maior salário: 6200.00 ==> Marklucio Penteado
Menor salário: 3800.00 ==> Lucycleide da Silva
Folha de pagamento: 19000.00
Salário médio: 4750.00
Desvio padrão dos salários: 909.67

Diferença de cada salário em relação ao valor máximo de 8000:
Nome: Marklucio Penteado, Matrícula: 123, Diferença: 1800.00
Nome: Suelenita Perira, Matrícula: 124, Diferença: 3200.00
Nome: Luxemburgo Soares, Matrícula: 125, Diferença: 3800.00
Nome: Lucycleide da Silva, Matrícula: 126, Diferença: 4200.00
>>>
```

Um exemplo de saída do programa

```

'''
ControleSalarial.py
-----
Faz o cadastro de vários empregados e calcula: Valor da folha de pagamento, maior
salário, menor salário, salário médio, desvio padrão dos salários e diferença de
cada salário em relação ao valor máximo e o desvio padrão.
-----
'''

import math

class ClsEmpregado:
    def __init__(self, nome_emp, mat_emp, setor_emp, sal_emp):
        self.nome_emp = nome_emp
        self.mat_emp = mat_emp
        self.setor_emp = setor_emp
        self.sal_emp = sal_emp

#-----
def VerificarMatricula(matricula, listaEmpregados):
    for Empregado in listaEmpregados:
        if (Empregado.mat_emp == matricula):
            return True #matrícula já existe
    return False

#=====
#Programa principal
numEmpregados = abs(int(input("Entre com o número de empregados: ")))
LstEmp = [] #cria lista em vazio para conter os empregados
somaSalarios = 0

#Registro dos empregados
for _ in range(numEmpregados):
    nome = input("Entre com o nome do empregado: ")

    while(True):
        matricula = int(input(f"Entre com o número da matrícula de {nome}: "))
        if not(VerificarMatricula(matricula, LstEmp)):
            break
        else:
            print("Matrícula já existe. Por favor, digite outra.")

    setor = input(f"Entre com o setor do empregado {nome}: ")
    salario = float(input(f"Entre com o salário de {nome}: "))
    Empregado = ClsEmpregado(nome, matricula, setor, salario) #cria instância
    LstEmp.append(Empregado)
    somaSalarios += salario
    print()
print("-----")

#Cálculos
maiorSal = max(Empregado.sal_emp for Empregado in LstEmp)
menorSal = min(Empregado.sal_emp for Empregado in LstEmp)
empregadosComMaiorSal = (
    [Empregado.nome_emp for Empregado in LstEmp
     if Empregado.sal_emp == maiorSal]
)

```

```

if(empregadosComMaiorSal):
    nomeMaiorSal = empregadosComMaiorSal[0]
else:
    nomeMaiorSal = "Desconhecido"
    empregadosComMenorSal = (
        [Empregado.nome_emp for Empregado in LstEmp if Empregado.sal_emp == menorSal]
    )
if(empregadosComMenorSal):
    nomeMenorSal = empregadosComMenorSal[0]
else:
    nomeMenorSal = "Desconhecido"

salMedio = somaSalarios / numEmpregados

#Cálculo do desvio padrão
variancia = sum((Empregado.sal_emp-salMedio)**2 for Empregado in LstEmp)/umEmpregados

desvioPadrao = math.sqrt(variancia)

salMax = 8000 #salário máximo estabelecido pela empresa

#Resultados
print(f"\nMaior salário: {maiorSal:.2f} ==> {nomeMaiorSal}")
print(f"Menor salário: {menorSal:.2f} ==> {nomeMenorSal}")
print(f"Folha de pagamento: {somaSalarios:.2f}")
print(f"Salário médio: {salMedio:.2f}")
print(f"Desvio padrão dos salários: {desvioPadrao:.2f}")
print()

#Diferença de cada salário em relação ao máximo
print("Diferença de cada salário em relação ao valor máximo de 8000:")
for Empregado in LstEmp:
    diferenca = salMax - Empregado.sal_emp
    print(f"Nome: {Empregado.nome_emp}, Matrícula: {Empregado.mat_emp}, "
          f"Diferença: {diferenca:.2f}")
#Fim do programa " ControleSalarial" -----

```