Enunciado Detalhado — Sistema de Gerenciamento de Funcionários e Folha de Pagamento

6 Objetivo

Desenvolver um sistema orientado a objetos em Python que gerencie os dados de funcionários de uma empresa e calcule a folha de pagamento mensal, utilizando herança, polimorfismo e persistência de dados em arquivos JSON.

Descrição Geral

O sistema deverá possuir dois perfis de usuário:

- **Gerente**: pode cadastrar, editar e remover funcionários, além de visualizar a folha de pagamento completa.
- **Funcionário**: pode visualizar apenas seu **próprio contracheque**, com informações de salário base, descontos e salário líquido.

O sistema deve salvar todas as informações em **arquivos** .json, garantindo que os dados permaneçam salvos mesmo após encerrar o programa.

S Estrutura Recomendada do Projeto

```
projeto_folha_pagamento/

main.py  # Arquivo principal com o menu
classes/
pessoa.py  # Classe base Pessoa
funcionario.py  # Classe Funcionario (herda de Pessoa)
gerente.py  # Classe Gerente (herda de
Funcionario)
folha_pagamento.py  # Classe responsável pelos cálculos e
relatórios

dados/
funcionarios.json  # Armazena dados dos funcionários
usuarios.json  # Armazena dados de login

(gerente/funcionário)

utils/
manipulador_json.py  # Funções auxiliares de
leitura/escrita JSON
```

& Classes e Herança

Representa uma pessoa genérica do sistema.

Atributos:

```
nome: strcpf: stridade: int
```

Métodos:

- __init__: inicializa os atributos.
- str : retorna uma representação textual da pessoa.
- to_dict: retorna um dicionário com os dados (para salvar em JSON).

Classe Funcionario (Herda de Pessoa)

Representa um funcionário da empresa.

Atributos adicionais:

```
cargo: str
salario_base: float
descontos: float
tipo contratacao: str (ex: "CLT", "Horista", "Comissionado")
```

Métodos:

- init : chama o construtor da classe Pessoa com super().
- calcular salario: calcula o salário líquido (salário base descontos).
- to dict: sobrescreve o método da classe base para incluir os novos campos.
- mostrar contracheque: exibe informações do salário líquido e descontos.

Sobrescrita: Caso haja diferentes tipos de funcionários (ex: horistas e comissionados), o método calcular_salario deve ser sobrescrito para ajustar o cálculo conforme o tipo de contratação.

Classe Gerente (Herda de Funcionario)

Representa o gerente do sistema, com permissões administrativas.

Atributos adicionais:

• senha: str (para login no menu)

Métodos:

• cadastrar_funcionario: adiciona um novo funcionário no arquivo JSON.

- editar funcionario: altera informações de um funcionário.
- remover funcionario: exclui um funcionário do arquivo.
- listar funcionarios: exibe todos os funcionários cadastrados.
- gerar_folha_pagamento: percorre todos os funcionários e soma os salários líquidos.

Esses métodos utilizarão a **classe auxiliar ManipuladorJSON** para ler e escrever os arquivos .json.

Classe FolhaPagamento

Responsável por gerar relatórios de pagamento e estatísticas.

Métodos:

- calcular total folha: soma o total de salários líquidos da empresa.
- gerar_relatorio_json: cria um relatório da folha de pagamento e salva em .json.
- mostrar resumo: mostra o total de funcionários e o valor total da folha.

Manipulação de Arquivos JSON

Crie o módulo manipulador json.py com as seguintes funções:

- ler arquivo json(caminho: str) -> list | dict
- salvar_arquivo_json(caminho: str, dados: list | dict)

Essas funções serão usadas em todo o sistema para persistência dos dados.

Menu Principal (main.py)

O menu será responsável por interagir com o usuário via terminal.

Fluxo sugerido:

- 1. Solicitar login (usuário e senha).
- 2. Identificar se é Gerente ou Funcionário.
- 3. Exibir o menu correspondente:

Menu Gerente:

- 1 Cadastrar Funcionário
- 2 Editar Funcionário

- 3 Remover Funcionário
- 4 Listar Funcionários
- 5 Gerar Folha de Pagamento
- 6 Sair

Menu Funcionário:

- 1 Consultar Contracheque
 2 Sair

Todas as interações devem usar os métodos das classes e os arquivos .json.