



Jorge Dantas



Mariana Lima



Marianna Rossi



Mohara
Nascimento

OBJETIVOS

1

Criar arquivo JSON, a cada ano, com informações dos funcionários

Admissão

Promoção

Demissão

Festa

2

Criar um arquivo log/txt com histórico de todas as demissões

3

Identificar penetras nas festas e criar arquivo de penetras

Criar arquivo JSON de informações

Resultado Planejado

```
"ID16": {  
  "Nome": "Vinicius Santos",  
  "Idade": "29",  
  "Cargo": "Gerente de Marketing",  
  "Salário": "8000.00",  
  "Email": "vinicius.santos@example.com",  
  "Data de Admissão": "2021-06-10",  
  "Subordinados": "",  
  "Promoção": {  
    "Antigo Cargo": "Desenvolvedor Pleno",  
    "Antigo Salário": "5500.00"  
  },  
  "Demissão": {  
    "Data de Demissão": "2022-06-05",  
    "Razão da Demissão": "Desempenho insatisfatório"  
  },  
  "Festa": "Compareceu"  
},
```

Admissão

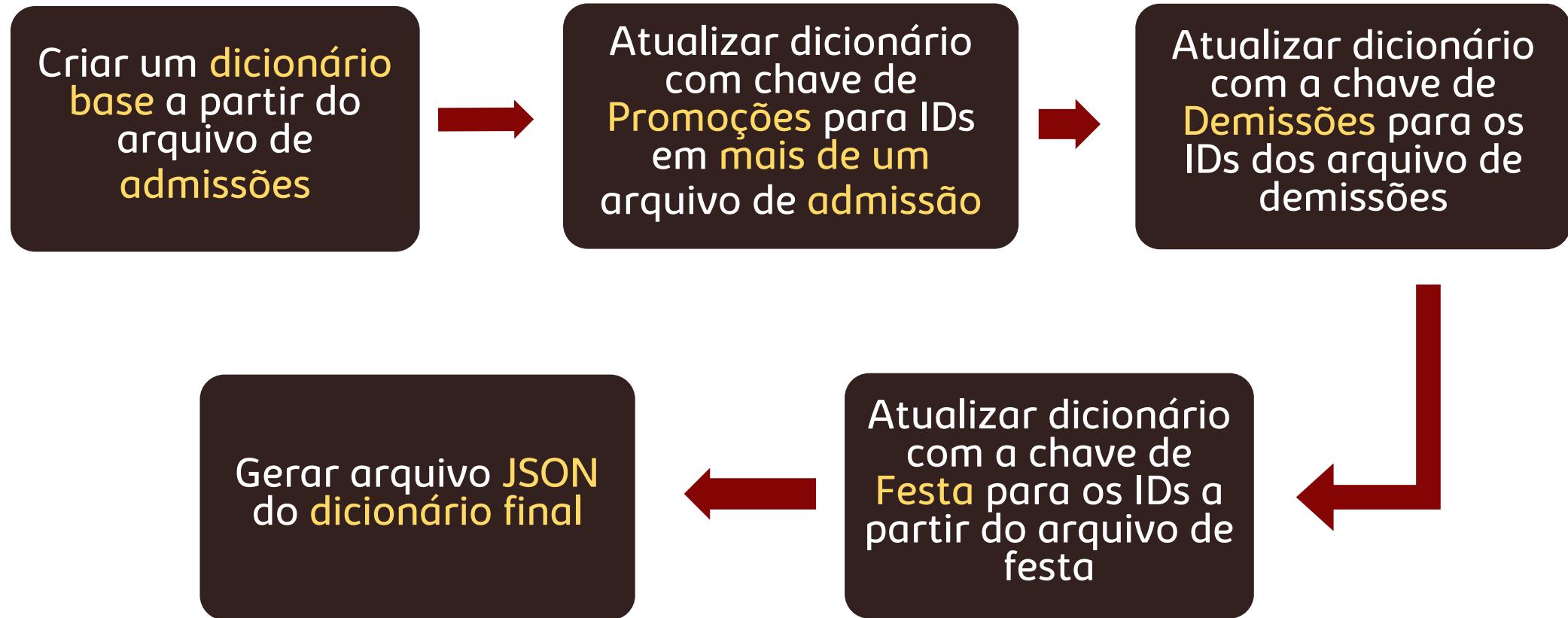
Promoção

Demissão

Festa

Criar arquivo JSON de informações

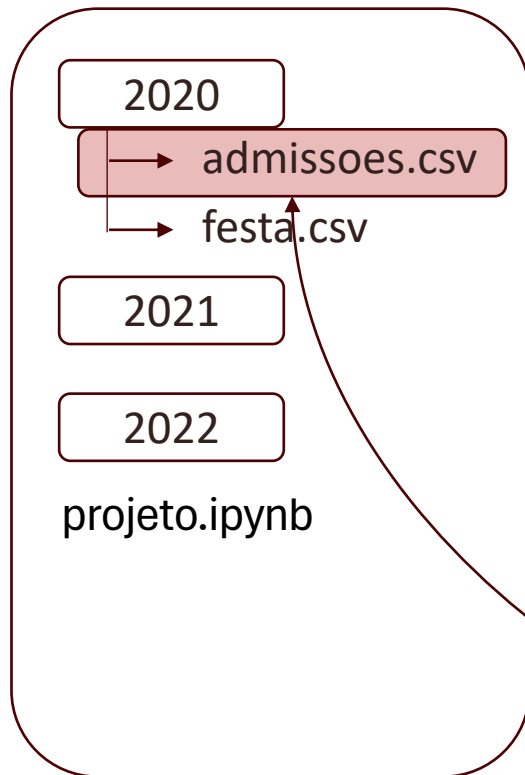
Workflow



Criar arquivo JSON de informações

1

Criar um dicionário base a partir de um arquivo



```
# Função que transforma um arquivo em dicionário
def dicionario_de_arquivo(ano:int, nome_do_arquivo:str):
    dicionario = {}
    arquivo = str(str(ano)+'/'+nome_do_arquivo)

    if not(os.path.exists(arquivo)):
        return dicionario

    lista = [linha for linha in (csv.reader(open(arquivo, 'r',encoding='cp1252'), delimiter=',', lineterminator='\n'))]
    dicionario = {chave[0]:{lista[0][i]:chave[i] for i in range(1,len(lista[0]))} for chave in lista[1:]}

    return dicionario
```

dicionario_base = dicionario_de_arquivo(2020,'admissoes.csv')

```
arquivo = str(str(ano)+'/'+nome_do_arquivo)
```

arquivo= '2020/admissoes.csv'

Criar arquivo JSON de informações

1

Criar um dicionário base a partir de um arquivo

```
lista = [linha for linha in (csv.reader(open(arquivo, 'r', encoding='cp1252'), delimiter=',', lineterminator='\n'))]
```



```
ID,Nome,Idade,Cargo,Salário,Email,Data de Admissão,Subordinados
ID1,João Silva,32,Gerente de Projetos,8000.00,joao.silva@example.com,2020-04-15,"ID5,ID8"
ID2,Maria Santos,28,Analista de Marketing,4500.00,maria.santos@example.com,2020-02-10,""
ID3,Pedro Souza,35,Desenvolvedor Sênior,6500.00,pedro.souza@example.com,2020-09-22,""
ID4,Ana Oliveira,31,Analista Financeiro,5000.00,ana.oliveira@example.com,2020-07-05,""
ID5,Lucas Pereira,27,Desenvolvedor Júnior,3800.00,lucas.pereira@example.com,2020-11-30,""
ID6,Carolina Santos,29,Analista de Recursos Humanos,4800.00,carolina.santos@example.com,2020-08-18,""
ID7,André Almeida,33,Gerente de Vendas,7500.00,andre.almeida@example.com,2020-03-26,"ID10"
ID8,Fernanda Costa,26,Analista de Sistemas,4200.00,fernanda.costa@example.com,2020-05-19,""
ID9,Rafael Lima,30,Desenvolvedor Pleno,5800.00,rafael.lima@example.com,2020-09-10,""
ID10,Patrícia Fernandes,34,Analista de Marketing,4700.00,patricia.fernandes@example.com,2020-07-15,""
ID11,Gustavo Rodrigues,32,Desenvolvedor Sênior,7000.00,gustavo.rodrigues@example.com,2020-12-01,"ID6"
ID12,Laura Mendes,29,Analista Financeiro,5200.00,laura.mendes@example.com,2020-04-25,""
ID13,Rodrigo Costa,26,Desenvolvedor Júnior,4000.00,rodrigo.costa@example.com,2020-10-10,""
ID14,Mariana Almeida,31,Analista de Recursos Humanos,4900.00,mariana.almeida@example.com,2020-08-05,""
ID15,Camila Oliveira,27,Gerente de Vendas,7200.00,camila.oliveira@example.com,2020-03-12,"ID3"
```

```
[ 'ID', 'Nome', 'Idade', 'Cargo', 'Salário', 'Email', 'Data de Admissão', 'Subordinados' ]
[ 'ID1', 'João Silva', '32', 'Gerente de Projetos', '8000.00', 'joao.silva@example.com', '2020-04-15', 'ID5,ID8' ]
[ 'ID2', 'Maria Santos', '28', 'Analista de Marketing', '4500.00', 'maria.santos@example.com', '2020-02-10', '' ]
[ 'ID3', 'Pedro Souza', '35', 'Desenvolvedor Sênior', '6500.00', 'pedro.souza@example.com', '2020-09-22', '' ]
[ 'ID4', 'Ana Oliveira', '31', 'Analista Financeiro', '5000.00', 'ana.oliveira@example.com', '2020-07-05', '' ]
[ 'ID5', 'Lucas Pereira', '27', 'Desenvolvedor Júnior', '3800.00', 'lucas.pereira@example.com', '2020-11-30', '' ]
[ 'ID6', 'Carolina Santos', '29', 'Analista de Recursos Humanos', '4800.00', 'carolina.santos@example.com', '2020-08-18', '' ]
[ 'ID7', 'André Almeida', '33', 'Gerente de Vendas', '7500.00', 'andre.almeida@example.com', '2020-03-26', 'ID10' ]
[ 'ID8', 'Fernanda Costa', '26', 'Analista de Sistemas', '4200.00', 'fernanda.costa@example.com', '2020-05-19', '' ]
[ 'ID9', 'Rafael Lima', '30', 'Desenvolvedor Pleno', '5800.00', 'rafael.lima@example.com', '2020-09-10', '' ]
[ 'ID10', 'Patrícia Fernandes', '34', 'Analista de Marketing', '4700.00', 'patricia.fernandes@example.com', '2020-07-15', '' ]
[ 'ID11', 'Gustavo Rodrigues', '32', 'Desenvolvedor Sênior', '7000.00', 'gustavo.rodrigues@example.com', '2020-12-01', 'ID6' ]
[ 'ID12', 'Laura Mendes', '29', 'Analista Financeiro', '5200.00', 'laura.mendes@example.com', '2020-04-25', '' ]
[ 'ID13', 'Rodrigo Costa', '26', 'Desenvolvedor Júnior', '4000.00', 'rodrigo.costa@example.com', '2020-10-10', '' ]
[ 'ID14', 'Mariana Almeida', '31', 'Analista de Recursos Humanos', '4900.00', 'mariana.almeida@example.com', '2020-08-05', '' ]
[ 'ID15', 'Camila Oliveira', '27', 'Gerente de Vendas', '7200.00', 'camila.oliveira@example.com', '2020-03-12', 'ID3' ]
```

Criar arquivo JSON de informações

1

Criar um dicionário base a partir de um arquivo

```
dicionario = {chave[0]:{lista[0][i]:chave[i] for i in range(1,len(lista[0]))} for chave in lista[1:]}
```

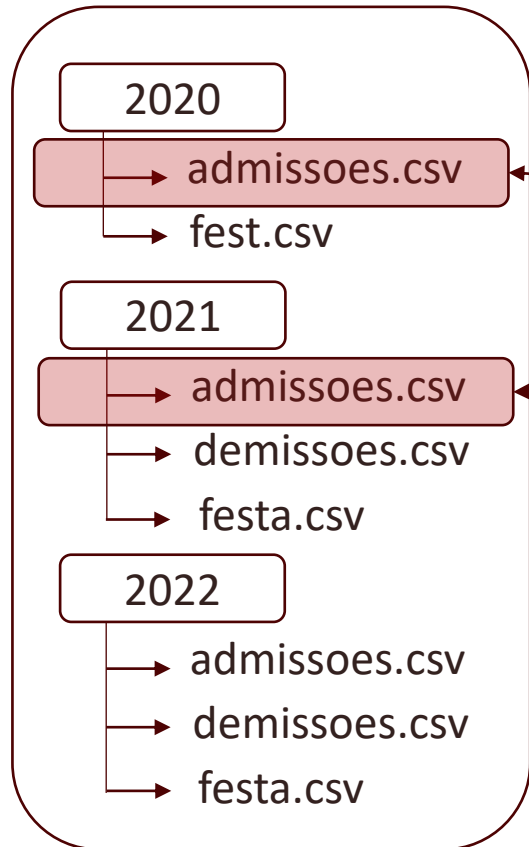
```
['ID', 'Nome', 'Idade', 'Cargo', 'Salário', 'Email', 'Data de Admissão', 'Subordinados']  
['ID1', 'João Silva', '32', 'Gerente de Projetos', '8000.00', 'joao.silva@example.com', '2020-04-15', 'ID5,ID8']  
['ID2', 'Maria Santos', '28', 'Analista de Marketing', '4500.00', 'maria.santos@example.com', '2020-02-10', '']  
['ID3', 'Pedro Souza', '35', 'Desenvolvedor Sênior', '6500.00', 'pedro.souza@example.com', '2020-09-22', '']  
['ID4', 'Ana Oliveira', '31', 'Analista Financeiro', '5000.00', 'ana.oliveira@example.com', '2020-07-05', '']  
['ID5', 'Lucas Pereira', '27', 'Desenvolvedor Júnior', '3800.00', 'lucas.pereira@example.com', '2020-11-30', '']  
['ID6', 'Carolina Santos', '29', 'Analista de Recursos Humanos', '4800.00', 'carolina.santos@example.com', '2020-08-18', '']  
['ID7', 'André Almeida', '33', 'Gerente de Vendas', '7500.00', 'andre.almeida@example.com', '2020-03-26', 'ID10']  
['ID8', 'Fernanda Costa', '26', 'Analista de Sistemas', '4200.00', 'fernanda.costa@example.com', '2020-05-19', '']  
['ID9', 'Rafael Lima', '30', 'Desenvolvedor Pleno', '5800.00', 'rafael.lima@example.com', '2020-09-10', '']  
['ID10', 'Patrícia Fernandes', '34', 'Analista de Marketing', '4700.00', 'patricia.fernandes@example.com', '2020-07-15', '']  
['ID11', 'Gustavo Rodrigues', '32', 'Desenvolvedor Sênior', '7000.00', 'gustavo.rodrigues@example.com', '2020-12-01', 'ID6']  
['ID12', 'Laura Mendes', '29', 'Analista Financeiro', '5200.00', 'laura.mendes@example.com', '2020-04-25', '']  
['ID13', 'Rodrigo Costa', '26', 'Desenvolvedor Júnior', '4000.00', 'rodrigo.costa@example.com', '2020-10-10', '']  
['ID14', 'Mariana Almeida', '31', 'Analista de Recursos Humanos', '4900.00', 'mariana.almeida@example.com', '2020-08-05', '']  
['ID15', 'Camila Oliveira', '27', 'Gerente de Vendas', '7000.00', 'camila.oliveira@example.com', '2020-03-12', 'ID3']
```

```
{  
  'ID1': {'Nome': 'João Silva', 'Idade': '32', 'Cargo': 'Gerente de Projetos', 'Salário': '8000.00', 'Email':  
    'joao.silva@example.com', 'Data de Admissão': '2020-04-15', 'Subordinados': 'ID5,ID8'},  
  'ID2': {'Nome': 'Maria Santos', 'Idade': '28', 'Cargo': 'Analista de Marketing', 'Salário': '4500.00', 'Email':  
    'maria.santos@example.com', 'Data de Admissão': '2020-02-10', 'Subordinados': ''},  
  'ID3': {'Nome': 'Pedro Souza', 'Idade': '35', 'Cargo': 'Desenvolvedor Sênior', 'Salário': '6500.00', 'Email':  
    'pedro.souza@example.com', 'Data de Admissão': '2020-09-22', 'Subordinados': ''},  
  'ID4': {'Nome': 'Ana Oliveira', 'Idade': '31', 'Cargo': 'Analista Financeiro', 'Salário': '5000.00', 'Email':  
    'ana.oliveira@example.com', 'Data de Admissão': '2020-07-05', 'Subordinados': ''},  
  'ID5': {'Nome': 'Lucas Pereira', 'Idade': '27', 'Cargo': 'Desenvolvedor Júnior', 'Salário': '3800.00', 'Email':  
    'lucas.pereira@example.com', 'Data de Admissão': '2020-11-30', 'Subordinados': ''},  
  'ID6': {'Nome': 'Carolina Santos', 'Idade': '29', 'Cargo': 'Analista de Recursos Humanos', 'Salário': '4800.00', 'Email':  
    'carolina.santos@example.com', 'Data de Admissão': '2020-08-18', 'Subordinados': ''},  
  'ID7': {'Nome': 'André Almeida', 'Idade': '33', 'Cargo': 'Gerente de Vendas', 'Salário': '7500.00', 'Email':  
    'andre.almeida@example.com', 'Data de Admissão': '2020-03-26', 'Subordinados': 'ID10'},  
  'ID8': {'Nome': 'Fernanda Costa', 'Idade': '26', 'Cargo': 'Analista de Sistemas', 'Salário': '4200.00', 'Email':  
    'fernanda.costa@example.com', 'Data de Admissão': '2020-05-19', 'Subordinados': ''},  
  'ID9': {'Nome': 'Rafael Lima', 'Idade': '30', 'Cargo': 'Desenvolvedor Pleno', 'Salário': '5800.00', 'Email':  
    'rafael.lima@example.com', 'Data de Admissão': '2020-09-10', 'Subordinados': ''},  
  'ID10': {'Nome': 'Patrícia Fernandes', 'Idade': '34', 'Cargo': 'Analista de Marketing', 'Salário': '4700.00', 'Email':  
    'patricia.fernandes@example.com', 'Data de Admissão': '2020-07-15', 'Subordinados': ''},  
  'ID11': {'Nome': 'Gustavo Rodrigues', 'Idade': '32', 'Cargo': 'Desenvolvedor Sênior', 'Salário': '7000.00', 'Email':  
    'gustavo.rodrigues@example.com', 'Data de Admissão': '2020-12-01', 'Subordinados': 'ID6'},  
  'ID12': {'Nome': 'Laura Mendes', 'Idade': '29', 'Cargo': 'Analista Financeiro', 'Salário': '5200.00', 'Email':  
    'laura.mendes@example.com', 'Data de Admissão': '2020-04-25', 'Subordinados': ''},  
  'ID13': {'Nome': 'Rodrigo Costa', 'Idade': '26', 'Cargo': 'Desenvolvedor Júnior', 'Salário': '4000.00', 'Email':  
    'rodrigo.costa@example.com', 'Data de Admissão': '2020-10-10', 'Subordinados': ''},  
  'ID14': {'Nome': 'Mariana Almeida', 'Idade': '31', 'Cargo': 'Analista de Recursos Humanos', 'Salário': '4900.00', 'Email':  
    'mariana.almeida@example.com', 'Data de Admissão': '2020-08-05', 'Subordinados': ''},  
  'ID15': {'Nome': 'Camila Oliveira', 'Idade': '27', 'Cargo': 'Gerente de Vendas', 'Salário': '7000.00', 'Email':  
    'camila.oliveira@example.com', 'Data de Admissão': '2020-03-12', 'Subordinados': 'ID3'}  
}
```


Criar arquivo JSON de informações

2

Atualizar dicionário base com chave Promoção



```
# Função que adiciona a chave de "Promoção" para um dicionário base e retorna o dicionário atualizado
def adicionar_promocoes(ano:int, dicionario:dict):

    ano_min = 2020 # ano inicial do banco de dados

    for id in dicionario:
        ano_aux = ano-1
        while ano_aux ≥ ano_min:
            dicionario_aux = dicionario_de_arquivo(ano_aux,'admissoes.csv')
            if id in dicionario_aux:
                if dicionario[id].get('Cargo') == dicionario_aux[id].get('Cargo'):
                    a = 'Funcionário não promovido, apenas demitido'
                else:
                    {dicionario[id].update({'Promoção':{'Antigo Cargo':dicionario_aux[id].get('Cargo'),'Antigo
Salário':dicionario_aux[id].get('Salário')}})}
            ano_aux -= 1
    return dicionario
```

dicionario_base = dicionario_de_arquivo(2022,'admissoes.csv')

adicionar_promocoes(2022,dicionario_base)

Criar arquivo JSON de informações

2

Atualizar dicionário base com chave Promoção

```
{dicionario[id].update({'Promoção':{'Antigo Cargo':dicionario_aux[id].get('Cargo'),'Antigo Salário':dicionario_aux[id].get('Salário')}})}
```

2021

```
{'ID16': {'Nome': 'Vinicius Santos',  
  'Idade': '28',  
  'Cargo': 'Desenvolvedor Pleno',  
  'Salário': '5500.00',  
  'Email': 'vinicius.santos@example.com',  
  'Data de Admissão': '2021-06-10',  
  'Subordinados': ''},
```

2020

```
'ID6': {'Nome': 'Carolina Santos',  
  'Idade': '29',  
  'Cargo': 'Analista de Recursos Humanos',  
  'Salário': '4800.00',  
  'Email': 'carolina.santos@example.com',  
  'Data de Admissão': '2020-08-18',  
  'Subordinados': ''},
```

2022

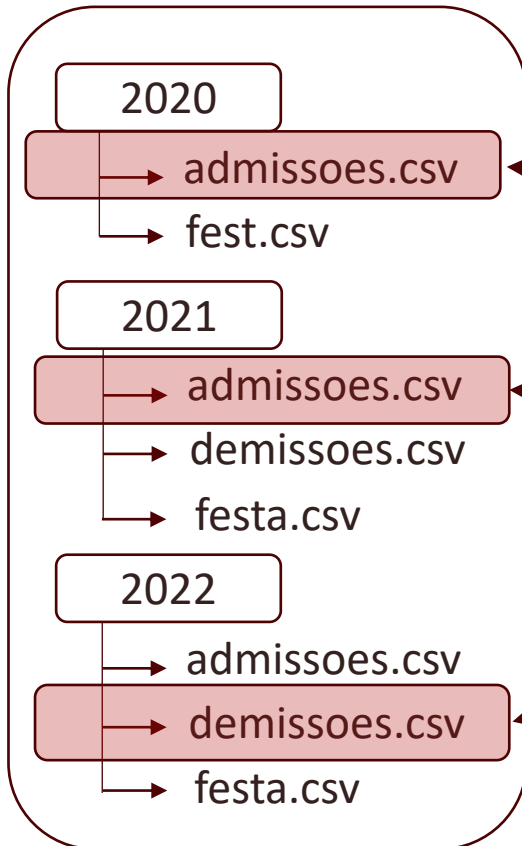
```
'ID16': {'Nome': 'Vinicius Santos',  
  'Idade': '29',  
  'Cargo': 'Gerente de Marketing',  
  'Salário': '8000.00',  
  'Email': 'vinicius.santos@example.com',  
  'Data de Admissão': '2021-06-10',  
  'Subordinados': '',  
  'Promoção': {'Antigo Cargo': 'Desenvolvedor Pleno',  
    'Antigo Salário': '5500.00'}},
```

```
{'ID6': {'Nome': 'Carolina Santos',  
  'Idade': '29',  
  'Cargo': 'Gerente de Projetos',  
  'Salário': '9000.00',  
  'Email': 'carolina.santos@example.com',  
  'Data de Admissão': '2020-08-18',  
  'Subordinados': 'ID11,ID14',  
  'Promoção': {'Antigo Cargo': 'Analista de Recursos Humanos',  
    'Antigo Salário': '4800.00'}},
```

Criar arquivo JSON de informações

3

Atualizar dicionário base com chave Demissão



```
# Função que adiciona a chave de "Demissão" para um dicionário base e retorna o dicionário atualizado
def adicionar_demissoes(ano:int, dicionario:dict):

    dicionario_de_demissoes = dicionario_de_arquivo(ano,'demissoes.csv') # criando dicionario de demissoes para usar como base

    if dicionario_de_demissoes == {}:
        return dicionario

    ano_min = 2020 # ano inicial do banco de dados, será usado para que as iterações ocorram do ano inserido até o ano_min

    for id in dicionario_de_demissoes:
        if id in dicionario:
            {dicionario[id_fun].update({'Demissão':{'Data de Demissão':dicionario_de_demissoes[id_dem].get('Data de Demissão'),'Razão da Demissão':dicionario_de_demissoes[id_dem].get('Razão da Demissão')}}) for id_fun in dicionario for id_dem in dicionario_de_demissoes if id_fun==id_dem}
        else:
            ano_aux = ano-1
            while ano_aux ≥ ano_min:
                dicionario_aux = dicionario_de_arquivo(ano_aux,'admissoes.csv')
                if id in dicionario_aux:
                    dicionario[id] = dicionario_aux[id]
                    {dicionario[id_fun].update({'Demissão':{'Data de Demissão':dicionario_de_demissoes[id_dem].get('Data de Demissão'),'Razão da Demissão':dicionario_de_demissoes[id_dem].get('Razão da Demissão')}}) for id_fun in dicionario for id_dem in dicionario_de_demissoes if id_fun==id_dem}
                ano_aux -= 1

    return dicionario
```

dicionario_base = dicionario_de_arquivo(2022,'admissoes.csv')

adicionar_promocoes(2022,dicionario_base) → adicionar_demissoes(2022,dicionario_base)

Criar arquivo JSON de informações

3

Atualizar dicionário base com chave Demissão

```
for id in dicionario_de_demissoes:  
    if id in dicionario:
```

2022

```
{'ID16': {'Nome': 'Vinicius Santos',  
  'Idade': '29',  
  'Cargo': 'Gerente de Marketing',  
  'Salário': '8000.00',  
  'Email': 'vinicius.santos@example.com',  
  'Data de Admissão': '2021-06-10',  
  'Subordinados': '',  
  'Promoção': {'Antigo Cargo': 'Desenvolvedor Pleno',  
    'Antigo Salário': '5500.00'},  
  'Demissão': {'Data de Demissão': '2022-06-05',  
    'Razão da Demissão': 'Desempenho insatisfatório'}}},
```

```
dicionario[id] = dicionario_aux[id]  
{dicionario[id_fun].update({'Demissão': {'Data de Demissão': dicionario_de_demissoes[id_dem].get('Data de  
Demissão'), 'Razão da Demissão': dicionario_de_demissoes[id_dem].get('Razão da Demissão')}}) for id_fun in dicionario  
for id_dem in dicionario_de_demissoes if id_fun==id_dem}
```

2021

```
{'ID18': {'Nome': 'Thiago Fernandes',  
  'Idade': '35',  
  'Cargo': 'Desenvolvedor Sênior',  
  'Salário': '6800.00',  
  'Email': 'thiago.fernandes@example.com',  
  'Data de Admissão': '2021-08-15',  
  'Subordinados': ''},
```

Demissão 2022

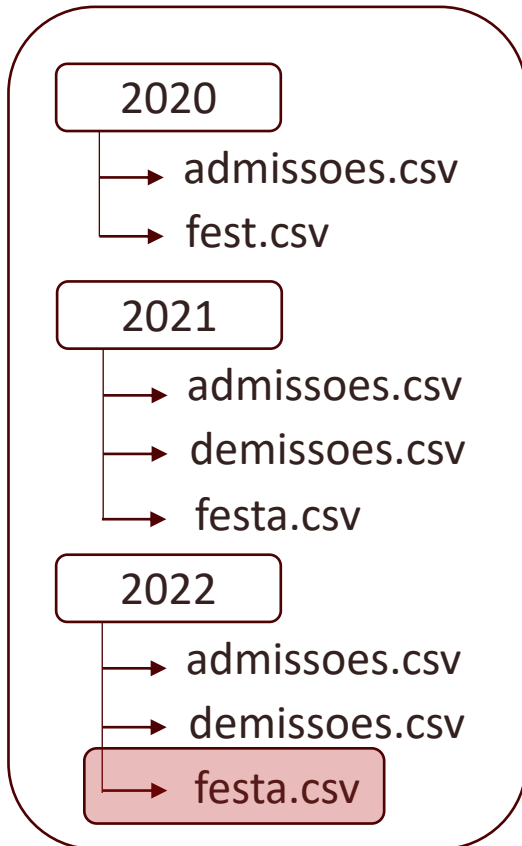
```
{'ID18': {'Nome': 'Thiago Fernandes',  
  'Idade': '35',  
  'Cargo': 'Desenvolvedor Sênior',  
  'Salário': '6800.00',  
  'Email': 'thiago.fernandes@example.com',  
  'Data de Demissão': '2022-09-10',  
  'Razão da Demissão': 'Mudança de carreira'},
```

```
{'ID18': {'Nome': 'Thiago Fernandes',  
  'Idade': '35',  
  'Cargo': 'Desenvolvedor Sênior',  
  'Salário': '6800.00',  
  'Email': 'thiago.fernandes@example.com',  
  'Data de Demissão': '2022-09-10',  
  'Razão da Demissão': 'Mudança de carreira',  
  'Demissão': {'Data de Demissão': '2022-09-10',  
    'Razão da Demissão': 'Mudança de carreira'}}},
```

Criar arquivo JSON de informações

4

Atualizar dicionário base com chave Festa



```
# Função que atualiza o dicionario base informando se o funcionário compareceu ou não à festa anual
def adicionar_festa(ano:int, dicionario:dict):

    dicionario_de_festa = dicionario_de_arquivo(ano,'festa.csv') # criando dicionario de demissoes para usar como base

    if dicionario_de_festa == {}:
        return dicionario

    for id in dicionario:
        if id in dicionario_de_festa:
            if dicionario_de_festa[id].get('Nome') == dicionario[id].get('Nome') and dicionario_de_festa[id].get('Idade') ==
dicionario[id].get('Idade'):
                {dicionario[id].update({'Festa':'Compareceu'})}
            else:
                {dicionario[id].update({'Festa':'Não Compareceu'})}
        else:
            {dicionario[id].update({'Festa':'Não Compareceu'})}

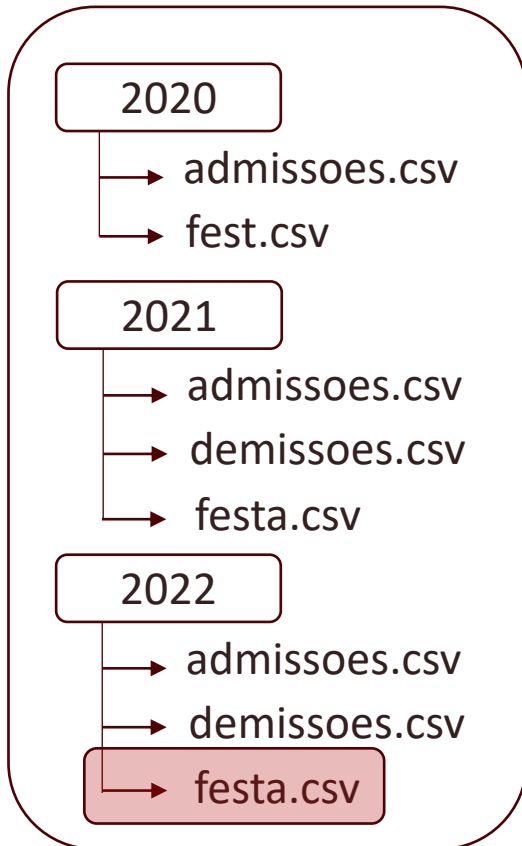
    return dicionario
```

```
dicionario_base = dicionario_de_arquivo(2022,'admissoes.csv')
adicionar_promocoos(2022,dicionario_base)
adicionar_demissoes(2022,dicionario_base)
adicionar_festa(2022,dicionario_base)
```

Criar arquivo JSON de informações

5

Atualizar dicionário base com chave Festa



```
# Função que junta todas as informações de um ano em um único dicionário
def gerar_dicionario_de_informacoes(ano:int):

    dicionario_ = dicionario_de_arquivo(ano,'admissoes.csv')
    adicionar_promocoes(ano,dicionario_)
    adicionar_demissoes(ano,dicionario_)
    adicionar_festa(ano,dicionario_)

    return dicionario_
```

```
dicionario_base = gerar_dicionário_de_informacoes(2022)
```

```
# Função que gera o arquivo JSON a partir de um dicionário
def gerar_arquivo_json(ano, dicionario:dict):

    nome_do_arquivo = (f'informacoes{ano}.json')
    with open(nome_do_arquivo,'w',encoding='utf8') as f:
        json.dump(dicionario, f, ensure_ascii=False)
```

```
gerar_arquivo_json(2022,dicionário_base)
```

Criar arquivo JSON de informações

6

Funcionamento das funções no menu final

```
-----Bem-vindo(a) ao gerenciador de dados da empresa-----  
1. Gerar arquivo de informações para um ano específico  
2. Gerar arquivo de informações para todos os anos  
3. Gerar arquivo com histórico de demissões  
4. Identificar penetras na festa de determinado ano  
5. Histórico de penetras nas festas  
6. Sair
```

Opção 1

```
lista_anos = [2020,2021,2022]  
  
opcao = input('Digite a opção escolhida: ')  
while opcao != '6':  
    if opcao == '1':  
        ano = int(input('Digite o ano que deseja gerar o arquivo (2020, 2021 ou 2022): '))  
  
        if not(ano in lista_anos):  
            raise Exception(f'Ano inválido. Digite um ano entre {lista_anos[0]} e {lista_anos[-1]}')  
        else:  
            dicionario_final = gerar_dicionario_de_informacoes(ano)  
            gerar_arquivo_json(ano,dicionario_final)
```

Criar arquivo JSON de informações

6

Funcionamento das funções no menu final

```
-----Bem-vindo(a) ao gerenciador de dados da empresa-----  
1. Gerar arquivo de informações para um ano específico  
2. Gerar arquivo de informações para todos os anos  
3. Gerar arquivo com histórico de demissões  
4. Identificar penetras na festa de determinado ano  
5. Histórico de penetras nas festas  
6. Sair
```

Opção 2

```
lista_anos = [2020,2021,2022]
```

```
elif opcao == '2':  
    dicionario_todos = {}  
    for ano in lista_anos:  
        dicionario_todos[ano]=gerar_dicionario_de_informacoes(ano)  
    gerar_arquivo_json('_todos',dicionario_todos)
```


Criar arquivo log/txt de demissões

Resultado Planejado

```
2021-06-05 Funcionário(a) Patrícia Fernandes de ID10 demitido(a) por Desempenho insatisfatório
2021-07-20 Funcionário(a) Fernanda Costa de ID8 demitido(a) por Mudança de carreira
2021-09-10 Funcionário(a) Lucas Pereira de ID5 demitido(a) por Insatisfação com o cargo
2022-06-05 Funcionário(a) Vinicius Santos de ID16 demitido(a) por Desempenho insatisfatório
2022-07-20 Funcionário(a) Gustavo Rodrigues de ID23 demitido(a) por Redução de equipe
2022-09-10 Funcionário(a) Thiago Fernandes de ID18 demitido(a) por Mudança de carreira
```

Criar arquivo log/txt de demissões

1 Ordenar as datas com auxílio de listas e do datetime

```
# Função que gera o arquivo log de demissões
def historico_de_demissoes(ano_inicial, ano_final):
    ano_min = ano_inicial
    ano_max = ano_final
    ano_aux = ano_min
    lista_datas = []

    # Parte 1 - Pegar as datas e salvar em uma lista ordenada (da data mais antiga para mais atual)
    while ano_aux <= ano_max:
        dicionario = dicionario_de_arquivo(ano_aux, 'demissoes.csv')
        for id in dicionario:
            data = datetime.strptime(dicionario[id].get('Data de Demissão'), '%Y-%m-%d')
            lista_datas.append(data)
        ano_aux += 1

    lista_datas.sort() # ordenar essa lista de datas

    lista_d_strings = [data.strftime('%Y-%m-%d') for data in lista_datas] # salvar a lista ordenada em
    strings no formato do nosso dicionario
```

```
[datetime.datetime(2021, 6, 5, 0, 0),
 datetime.datetime(2021, 7, 20, 0, 0),
 datetime.datetime(2021, 9, 10, 0, 0),
 datetime.datetime(2022, 6, 5, 0, 0),
 datetime.datetime(2022, 7, 20, 0, 0),
 datetime.datetime(2022, 9, 10, 0, 0)]
```

```
['2021-06-05',
 '2021-07-20',
 '2021-09-10',
 '2022-06-05',
 '2022-07-20',
 '2022-09-10']
```

Criar arquivo log/txt de demissões

2


Percorrer os dicionários de demissões e salvar as informações no arquivo

```
# Parte 2 - Salvar as informações dos funcionarios já formatados pro arquivo log (da data mais antiga para mais atual)
lista_demissoes = []

for data in lista_d_strings:
    ano_aux = ano_min
    while ano_aux ≤ ano_max:
        dicionario = dicionario_de_arquivo(ano_aux, 'demissoes.csv')
        for id in dicionario:
            if data == dicionario[id].get('Data de Demissão'):
                nome = dicionario[id].get('Nome')
                razao = dicionario[id].get('Razão da Demissão')
                informacao_log = f'{data} Funcionário(a) {nome} de {id} demitido(a) por {razao}'
                lista_demissoes.append(informacao_log)
            ano_aux += 1

with open('historico_de_demissoes.txt', 'w', encoding='utf8') as arquivo:
    arquivo.write("\n".join(lista_demissoes))

return arquivo
```



```
2021-06-05 Funcionário(a) Patrícia Fernandes de ID10 demitido(a) por Desempenho insatisfatório
2021-07-20 Funcionário(a) Fernanda Costa de ID8 demitido(a) por Mudança de carreira
2021-09-10 Funcionário(a) Lucas Pereira de ID5 demitido(a) por Insatisfação com o cargo
2022-06-05 Funcionário(a) Vinicius Santos de ID16 demitido(a) por Desempenho insatisfatório
2022-07-20 Funcionário(a) Gustavo Rodrigues de ID23 demitido(a) por Redução de equipe
2022-09-10 Funcionário(a) Thiago Fernandes de ID18 demitido(a) por Mudança de carreira
```

Criar arquivo log/txt de demissões

3

Funcionamento da função no menu final

```
-----Bem-vindo(a) ao gerenciador de dados da empresa-----  
1. Gerar arquivo de informações para um ano específico  
2. Gerar arquivo de informações para todos os anos  
3. Gerar arquivo com histórico de demissões  
4. Identificar penetras na festa de determinado ano  
5. Histórico de penetras nas festas  
6. Sair
```

Opção 3

```
lista_anos = [2020,2021,2022]
```

```
elif opcao == '3':  
    historico_de_demissoes(lista_anos[0],lista_anos[-1])
```

Criar arquivo(s) de penetras

Resultado Planejado

Penetras de determinado ano (ex. 2022)

```
[
  {
    "ID": "ID1",
    "Nome": "João Souza",
    "Idade": "32"
  },
  {
    "ID": "ID26",
    "Nome": "Natália Lima",
    "Idade": "25"
  },
  {
    "ID": "ID27",
    "Nome": "Gabriel Silva",
    "Idade": "26"
  }
]
```

Histórico de penetras de todos os anos

```
[[2020, [{"ID": "ID1", "Nome": "João Souza", "Idade": "32"}, {"ID": "ID10", "Nome": "Patrícia Fernanda", "Idade": "34"}, {"ID": "ID16", "Nome": "Vinicius Santos", "Idade": "29"}, {"ID": "ID26", "Nome": "Natália Lima", "Idade": "25"}, {"ID": "ID27", "Nome": "Gabriel Silva", "Idade": "26"}, {"ID": "ID28", "Nome": "Mariana Almeida", "Idade": "29"}, {"ID": "ID29", "Nome": "Vitor Oliveira", "Idade": "33"}, {"ID": "ID30", "Nome": "Júlia Rodrigues", "Idade": "27"}]], [2021, [{"ID": "ID1", "Nome": "João Souza", "Idade": "32"}, {"ID": "ID16", "Nome": "Vinicius Santos", "Idade": "29"}, {"ID": "ID26", "Nome": "Natália Lima", "Idade": "25"}, {"ID": "ID27", "Nome": "Gabriel Silva", "Idade": "26"}, {"ID": "ID28", "Nome": "Mariana Almeida", "Idade": "29"}, {"ID": "ID29", "Nome": "Vitor Oliveira", "Idade": "33"}, {"ID": "ID30", "Nome": "Júlia Rodrigues", "Idade": "27"}]], [2022, [{"ID": "ID1", "Nome": "João Souza", "Idade": "32"}, {"ID": "ID26", "Nome": "Natália Lima", "Idade": "25"}, {"ID": "ID27", "Nome": "Gabriel Silva", "Idade": "26"}]]]
```

Criar arquivo(s) de penetras

1

Funções auxiliares para validar participante, registrar penetra e gerar arquivo final

```
# Função que verifica se uma pessoa está autorizada a entrar no evento de confraternização
def validar_participante(dicionario: dict, id: str, nome: str, idade: int) → bool:
    if id in dicionario:
        return dicionario[id].get('Nome') == nome and dicionario[id].get('Idade') == idade
    return False
```

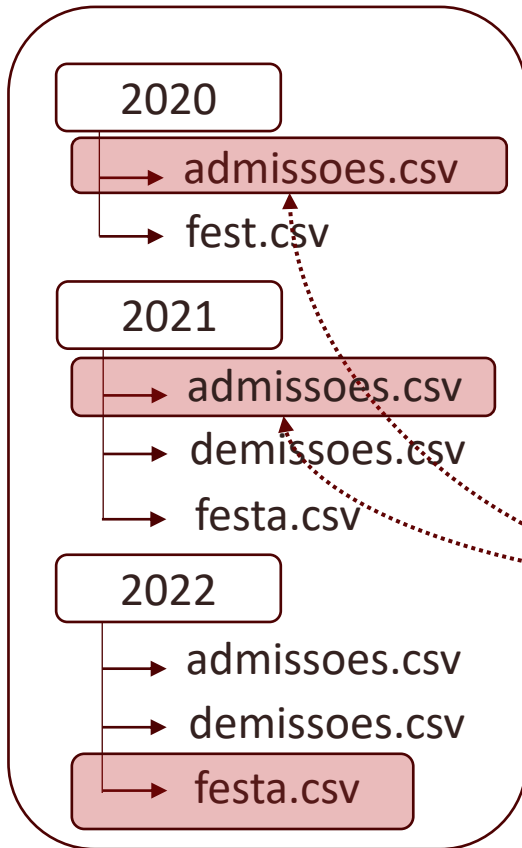
```
# Função que armazena informações sobre penetras em uma lista
def registrar_penetra(penetras: list, id: str, nome: str, idade: int):
    penetrante = {'ID': id, 'Nome': nome, 'Idade': idade}
    return penetras.append(penetrante)
```

```
# Salvar as informações dos penetras no arquivo JSON
def arquivo_penetras(lista_de_penetras: list, ano):
    nome_do_arquivo = (f'penetras{ano}.json')
    with open(nome_do_arquivo, 'w', encoding='utf8') as f:
        json.dump(lista_de_penetras, f, ensure_ascii=False)
```

Criar arquivo(s) de penetras

2

Verificar se as pessoas que querem entrar na festa são da empresa



```
# Função que gerencia os penetras da festa de confraternização verificando se a pessoa faz parte do cadastro
def gerenciar_penetras(ano:int):
    # Carregar informações dos penetras (se o arquivo existir)
    nome_do_arquivo_penetras = f'penetras{ano}.json'
    if os.path.exists(nome_do_arquivo_penetras):
        with open(nome_do_arquivo_penetras, 'r') as f:
            penetras = json.load(f)
    else:
        penetras = []

    dicionario_de_festa = dicionario_de_arquivo(ano, 'festa.csv')

    if dicionario_de_festa == {}:
        return penetras

    for id, participante in dicionario_de_festa.items():
        ano_min = 2020
        ano_max = ano
        id_esta_em_dict = False # Para controlar se a pessoa está em algum dicionário

        while ano_max >= ano_min:
            dicionario = dicionario_de_arquivo(ano_max, 'admissoes.csv')
            if id in dicionario:
                id_esta_em_dict = True
                if validar_participante(dicionario, id, participante.get('Nome'), participante.get('Idade')) == False:
                    registrar_penetra(penetras, id, participante.get('Nome'), participante.get('Idade'))
                break
            else:
                ano_max -= 1

        if id_esta_em_dict == False:
            registrar_penetra(penetras, id, participante.get('Nome'), participante.get('Idade'))

    return penetras
```


Criar arquivo(s) de penetras

3

Funcionamento da função no menu final

```
-----Bem-vindo(a) ao gerenciador de dados da empresa-----  
1. Gerar arquivo de informações para um ano específico  
2. Gerar arquivo de informações para todos os anos  
3. Gerar arquivo com histórico de demissões  
4. Identificar penetras na festa de determinado ano  
5. Histórico de penetras nas festas  
6. Sair
```

Opção 4 e Opção 5

```
lista_anos = [2020,2021,2022]
```

```
elif opcao == '4':  
    ano = int(input('Digite o ano do qual você deseja identificar os penetras (2020, 2021 ou 2022): '))  
    arquivo_penetras(gerenciar_penetras(ano),ano)  
elif opcao == '5':  
    lista_penetras = []  
    for ano in lista_anos:  
        lista_penetras.append([ano,gerenciar_penetras(ano)])  
    arquivo_penetras(lista_penetras,'_todos')
```

Outputs finais

Caso a pessoa opte por percorrer todas as opções, os outputs finais são:

```
≡ historico_de_demissoes.txt  
{ } informacoes_todos.json  
{ } informacoes2020.json  
{ } informacoes2021.json  
{ } informacoes2022.json  
{ } penetras_todos.json  
{ } penetras2020.json  
{ } penetras2021.json  
{ } penetras2022.json
```

Obrigado(a)!

github.com/moharanascimento/projeto-final-lpii-grupo-2-sc