

### \$\$\$ controle.alocacao\_equipamentos\_construcao\_civil

```
from src.util.gerais import imprimir_objetos
from src.util.data import Data
from src.entidades.empregado import Empregado, get_empregados,
inserir_empregado
from src.entidades.equipamento import Equipamento, get_equipamentos,
inserir_equipamento

def cadastrar_equipamentos():
    inserir_equipamento(Equipamento("Furadeira", "Ferramenta Elétrica", 20, False))
    inserir_equipamento(Equipamento("Lixadeira", "Ferramenta Elétrica", 18, True))
    inserir_equipamento(Equipamento("Betoneira", "Construção", 320, False))
    inserir_equipamento(Equipamento("Andaime", "Construção", 200, True))
    inserir_equipamento(Equipamento("Capacete de proteção", "Segurança", 1,
False))
    inserir_equipamento(Equipamento("Cinto de segurança", "Segurança", 3, False))
    inserir_equipamento(Equipamento("Carrinho de mão", "Transporte", 15, True))
    inserir_equipamento(Equipamento("Empilhadeira", "Transporte", 3500, True))
    inserir_equipamento(Equipamento("Trena", "Medição", 0.5, False))
    inserir_equipamento(Equipamento("Nível a laser", "Medição", 2, True))
    inserir_equipamento(Equipamento("Serra circular", "Corte", 12, False))
    inserir_equipamento(Equipamento("Esmerilhadeira", "Corte", 8, True))
    inserir_equipamento(Equipamento("Guincho", "Elevação", 500, True))
    inserir_equipamento(Equipamento("Talha manual", "Elevação", 50, False))

def cadastrar_empregados():
    inserir_empregado(Empregado("Construções Dourados Ltda.", "(67) 9999-8888",
"contato.dourados@gmail.com", "Rua dos Pedreiros, 123"))
    inserir_empregado(Empregado("Serviços de Telecomunicação", "(67) 8888-7777",
"tele.servicos@gmail.com", "Avenida da Obra, 456"))
    inserir_empregado(Empregado("Construção Civil", "(67) 7777-6666",
"contato.g3@gmail.com", "Rua da Inovação, 789"))

if __name__ == '__main__':
    print("\nAlocação de Equipamentos da Construção Civil")
    cadastrar_equipamentos()
    imprimir_objetos('Equipamento: nome, tipo, valor, disponível',
get_equipamentos())
    cadastrar_empregados()
    imprimir_objetos('Empregado: nome, telefone, email, endereço',
get_empregados())
```

### \$\$\$ entidades.empregado

```
empregados = []
```

```
def get_empregados(): return empregados
```

```
def inserir_empregado(empregado): empregados.append(empregado)
```

```
class Empregado:
```

```
    def __init__(self, nome, telefone, email, endereço):
```

```
        self.nome = nome
```

```
        self.telefone = telefone
```

```
        self.email = email
```

```
        self.endereço = endereço
```

```
    def __str__(self):
```

```
        formato = '{:<28} {:<16} {:<30} {:<30}'
```

```
        return formato.format(self.nome, self.telefone, self.email, self.endereço)
```

### \$\$\$ entidades.equipamento

```
equipamentos = []
```

```
class Equipamento:
```

```
    def __init__(self, nome, tipo, valor, disponivel):
```

```
        self.nome = nome
```

```
        self.tipo = tipo if tipo in ('Ferramenta Elétrica', 'Construção', 'Segurança',  
'Transporte', 'Medição', 'Corte', 'Elevação') else 'indefinido'
```

```
        self.valor = valor
```

```
        self.disponivel = disponivel
```

```
    def __str__(self):
```

```
        disponivel_str = 'disponível' if self.disponivel else "
```

```
        return f'{self.nome:<25} {self.tipo:<25} R$ {self.valor:<10.2f} {disponivel_str:<15}'
```

```
def get_equipamentos():
```

```
    return equipamentos
```

```
def inserir_equipamento(equipamento):
```

```
    equipamentos.append(equipamento)
```

### \$\$\$ util.data

```
from datetime import date
```

```
class Data:
```

```
    def __init__(self, dia, mês, ano):
```

```
        self.dia = dia
```

```
        self.mes = mês
```

```
        self.ano = ano
```

```
    def __str__(self):
```

```
        data_str = "
```

```
        if self.dia < 10:
```

```
            data_str = '0' + str(self.dia)
```

```
        else:
```

```
            data_str = str(self.dia)
```

```
        if self.mês < 10:
```

```
            data_str += "/0" + str(self.mês)
```

```
        else:
```

```
            data_str += "/" + str(self.mês)
```

```
        data_str += "/" + str(self.ano)
```

```
        return data_str
```

```
    def calcular_idade(self, data_referencia=None):
```

```
        if data_referencia is None:
```

```
            dia_atual_str, mês_atual_str, ano_atual_str =
```

```
date.today().strftime("%d/%m/%Y").split('/')
```

```
            dia_referencia, mês_referencia, ano_referencia = int(dia_atual_str),
```

```
int(mês_atual_str), int(ano_atual_str)
```

```
        else:
```

```
            dia_referencia, mês_referencia, ano_referencia = data_referencia.dia,  
data_referencia.mes, data_referencia.ano
```

```
        idade = ano_referencia - self.ano
```

```
        if mês_referencia < self.mês or (mês_referencia == self.mês and dia_referencia  
< self.dia):
```

```
            idade -= 1
```

```
        return idade
```

### **\$\$\$ util.gerais**

```
def compacta_nome(nome):
    partes_nome = nome.split(' ')
    nome_compactado = ""
    for parte_nome in partes_nome:
        nome_compactado += parte_nome if parte_nome not in ('de', 'da', 'das', 'do',
'dos') else "
    return nome_compactado
def imprimir_objetos(cabecalho, objetos):
    print('\n' + cabecalho)
    for indice, objeto in enumerate(objetos):
        formato = '{:<5} {}'
        print(formato.format(str(indice + 1) + ' -', str(objeto)))
```

Dourados, 22/08/2025

Rian Ruela Ribeiro