

Departamento de Engenharia de Sistemas e Computação

Implementação 1

Estrutura básica do programa, funções principais e dados

João Araujo

Algoritmos Computacionais I

Aula 31

Objetivo da aula

- Construir o **esqueleto** do programa: arquivos, funções principais e estruturas de dados
- Ter, ao final, uma versão que executa do início ao fim (mesmo que incompleta)
- Feedback contínuo do professor durante a aula

Princípio: desenvolvimento incremental

Escreva um pouco, **execute**, teste, corrija; só então avance. Nunca escreva 200 linhas antes de rodar pela primeira vez.

Roteiro de implementação

1. Crie a **pasta e os arquivos** previstos na especificação
2. Escreva os **esqueletos das funções** (nome, parâmetros, docstring, `pass`)
3. Defina a **estrutura de dados** central e crie dados de exemplo *no próprio código*
4. Implemente a **funcionalidade mais simples** de ponta a ponta (entrada $\rightarrow$ cálculo $\rightarrow$ saída)
5. Escreva o `main()` com o menu/fluxo geral
6. Teste, **integre** o trabalho dos integrantes e repita

Passo 2: esqueleto de funções

```python
def cadastra_insumo(estoque, nome, unidade, preco, qtd, minimo):
    """Acrescenta um insumo ao estoque (dicionário). Devolve True se criou."""
    pass

def baixa_estoque(estoque, nome, qtd):
    """Retira qtd do estoque. Levanta ValueError se faltar material."""
    pass

def custo_total(estoque):
    """Soma preco * qtd de todos os insumos."""
    pass

def insumos_em_alerta(estoque):
    """Lista os nomes com qtd menor ou igual ao mínimo."""
    pass
```

Nomes, parâmetros e *docstrings* já definem o **contrato** de cada função. Os integrantes podem trabalhar em funções diferentes.

Passo 3 e 4: dados e primeira funcionalidade

```python
def custo_total(estoque):
    """Soma preco * qtd de todos os insumos."""
    total = 0.0
    for dados in estoque.values():
        total += dados["preco"] * dados["qtd"]
    return total

def main():
    estoque = {
        "cimento": {"unidade": "saco", "preco": 38.5, "qtd": 120, "minimo": 30},
        "areia":   {"unidade": "m3",   "preco": 95.0, "qtd": 15,  "minimo": 10},
    }
    print(f"Custo total em estoque: R$ {custo_total(estoque):.2f}")

main()
```

Passo 5: o menu principal

```
def mostra_menu():
    print("\n1 - Cadastrar insumo\n2 - Baixar estoque")
    print("3 - Custo total\n4 - Alertas\n0 - Sair")

def main():
    estoque = {}
    while True:
        mostra_menu()
        opcao = input("Opção: ").strip()
        if opcao == "1":
            ...       # le dados e chama cadastra_insumo
        elif opcao == "3":
            print(f"Custo: R$ {custo_total(estoque):.2f}")
        elif opcao == "0":
            break
        else:
            print("Opção inválida.")
```

Cada opção do menu chama uma função: `main` só *coordena*.

Dividindo o trabalho em equipe

Boas práticas

- Cada integrante assume **funções ou módulos** distintos
- Combinem o **contrato** (parâmetros e retorno) antes de escrever
- Evitem editar o mesmo arquivo ao mesmo tempo
- Integrem *no mesmo dia*, não na véspera

Armadilhas

- Cada um com sua versão "do arquivo"
- Variáveis globais compartilhadas entre módulos
- Funções que leem do teclado *e* calculam (difíceis de testar)
- Nomes diferentes para a mesma coisa

Estruturas de dados: escolha consciente

Preciso de...	**Use**
uma coleção ordenada de itens iguais	lista
buscar por nome/código	dicionário
registro fixo (coordenada, data)	tupla
muitas operações numéricas	`np.array`
tabela com colunas nomeadas	lista de dicionários ou `DataFrame`
grade/tabuleiro	lista de listas (matriz)

"Uma estrutura de dados mal escolhida pode arruinar o melhor algoritmo." (Cap. 9.1)

Atividade: ao final desta aula

Cada grupo deve mostrar ao professor:

- A **estrutura de pastas/arquivos** criada
- Os **esqueletos das funções** com docstrings
- Pelo menos **uma funcionalidade** completa funcionando (entrada $\rightarrow$ saída)
- O `main()` com o fluxo geral (menu ou sequência de passos)

Feedback do professor

Verificará: funções coesas? Nomes claros? Dados bem escolhidos? O programa *executa*?

Checklist de implementação 1

- ☐ O programa roda sem erro do início ao fim?
- ☐ Cada função tem nome, parâmetros e docstring?
- ☐ Há uma função `main()` e o programa está dividido em módulos/funções?
- ☐ A estrutura de dados central está definida?
- ☐ Uma funcionalidade essencial está completa?
- ☐ O que falta está listado (para a Aula 32)?