

Ficha Rápida – Projeto 1: Fatorial de um Número

Objetivo do projeto

Criar um programa que recebe um número inteiro positivo **n** e imprime o seu **fatorial** ($n!$). Aplicar o método 5Q para estruturar o raciocínio e implementar a solução em Python usando laço **for**.

Conceito de fatorial

O fatorial de um número inteiro positivo **n** é o produto de todos os inteiros positivos de 1 até **n**. Ex.: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$. Pela propriedade comutativa, a ordem dos fatores não altera o produto.

Método 5Q (resumo aplicado)

- **1) Análise crítica:** precisamos calcular $n!$; confirmar que **n** é inteiro positivo.
- **2) Dados de entrada:** um número inteiro positivo **n**.
- **3) O que fazer com os dados:** multiplicar todos os inteiros de 1 até **n**.
- **4) Restrições:** **n** deve ser inteiro e $n \geq 1$ (na aula, tratamos positivos; $0! = 1$ pode ser considerado como extensão).
- **5) Sequência de passos (pseudocódigo):** receber **n** → validar → inicializar fatorial = 1 → percorrer **i** de 1 até **n** → fatorial = fatorial \times **i** → exibir resultado.

Pseudocódigo

```
ler n
se n é inteiro positivo:
    fatorial = 1
    para i de 1 até n:
        fatorial = fatorial * i
    imprimir "O fatorial de n é fatorial"
senão:
    imprimir "Informe apenas números inteiros positivos"
```

Implementação em Python (versão iterativa)

```
numero = int(input("Digite o fatorial que deseja calcular: "))

if numero > 0: # validação de positivo
    fatorial = 1
    for i in range(1, numero + 1):
        fatorial = fatorial * i
        print(f"{fatorial // i} x {i} = {fatorial}") # passo a passo (opcional)

    print(f"O fatorial de {numero} é {fatorial}")
else:
    print("Por favor, informe apenas números inteiros positivos")
```

Demonstração (exemplo com 5!)

Passo	Cálculo	Resultado parcial
1	1×1	1
2	1×2	2
3	2×3	6
4	6×4	24
5	24×5	120

Variações e extensões (para prática)

- Aceitar 0 como entrada e retornar 1 ($0! = 1$).
- Exibir apenas o resultado final (sem passos).
- Tratar entradas inválidas (texto, número negativo) com mensagens claras e nova solicitação.
- Comparar com uma versão recursiva (apenas para estudo; na aula usamos laço **for**).

Erros comuns (e como evitar)

- Não validar que o número é positivo: adicione uma verificação antes do cálculo.
- Esquecer de incluir o último valor no range: use `range(1, n+1)`.
- Multiplicação incorreta por iniciar fatorial em 0: inicie sempre em 1.
- Laço com limites errados (ex.: `range(n)` começa em 0 e não inclui `n`).

Ficha baseada no conteúdo do projeto Fatorial apresentado em aula. Use esta referência para revisar a lógica, o método 5Q e a implementação iterativa em Python.