

Tratamento de Exceções

Lista de Exercícios I

- 1) Solicite ao usuário o valor total da compra (permitindo centavos) e a quantidade de itens (inteiro); calcule o preço unitário dividindo o valor total pela quantidade e exiba o resultado com duas casas decimais, tratando adequadamente **entradas não numéricas, a quantidade igual a zero e quaisquer valores negativos informados para o total ou para a quantidade**.

Execuções simuladas

Caso 1 — dados válidos

Digite o valor total (R\$): 37.50

Digite a quantidade de itens: 3

Preço unitário: R\$ 12.50

Caso 2 — quantidade zero

Digite o valor total (R\$): 80

Digite a quantidade de itens: 0

Erro: a quantidade não pode ser zero.

Caso 3 — entrada não numérica (valor total)

Digite o valor total (R\$): trinta

Erro: digite números válidos.

Caso 4 — entrada não numérica (quantidade inteira esperada)

Digite o valor total (R\$): 50

Digite a quantidade de itens: 2.5

Erro: digite números válidos.

Caso 5 — valor negativo

Digite o valor total (R\$): -10

Digite a quantidade de itens: 2

Erro: valores negativos não são permitidos.

Caso 6 — quantidade negativa

Digite o valor total (R\$): 100

Digite a quantidade de itens: -4

Erro: valores negativos não são permitidos.

- 2) Considere a lista `["arroz", "feijão", "macarrão", "óleo"]`.

Peça ao usuário um índice e mostre o item correspondente.

Trate **entrada não numérica** e **índice fora da faixa**.

Execuções simuladas

Caso 1 — índice válido

Digite o índice do item: 2

Item selecionado: macarrão

Caso 2 — índice fora da faixa

Digite o índice do item: 10

Erro: índice inexistente na lista.

Caso 3 — entrada não numérica

Digite o índice do item: dois

Erro: você digitou um valor que não é número inteiro.

- 3) Tente abrir um arquivo chamado `dados.txt`, ler seu conteúdo e convertê-lo para número inteiro. Trate **arquivo inexistente** e **conteúdo não numérico**, exibindo mensagens adequadas.

Execuções simuladas

Caso 1 — arquivo existe e é numérico

Lendo 'dados.txt'...

Conteúdo: 123

Número lido: 123

Caso 2 — arquivo não existe

Lendo 'dados.txt'...

Erro: o arquivo 'dados.txt' não foi encontrado.

Caso 3 — conteúdo não numérico

Lendo 'dados.txt'...

Conteúdo: ABC

Erro: conteúdo do arquivo não é um número inteiro.

- 4) Solicite ao usuário dois números inteiros, calcule a divisão do primeiro pelo segundo e exiba o resultado apenas se nenhuma falha ocorrer, utilizando obrigatoriamente o bloco `else`; trate adequadamente a **entrada não numérica** e a **tentativa de divisão por zero**.

Execuções simuladas

Caso 1 — sem erro (executa o else)

Digite o numerador: 20

Digite o denominador: 4

Resultado: 5.0

Caso 2 — divisão por zero

Digite o numerador: 15

Digite o denominador: 0

Erro: não é possível dividir por zero.

Caso 3 — entrada não numérica

Digite o numerador: dez

Erro: você digitou um valor que não é número inteiro.

- 5) Crie uma função que abra um arquivo de texto, leia a primeira linha e exiba o conteúdo. Garanta que o arquivo será **sempre** fechado, mesmo se ocorrer erro de abertura ou leitura (use `finally`). Trate **arquivo inexistente** e **falha de leitura**.

Execuções simuladas

Caso 1 — arquivo encontrado

Abrindo 'notas.txt'...

Primeira linha: Aluno: Ana; Nota: 9.2

Fechando arquivo... (sempre executado)

Caso 2 — arquivo ausente

Abrindo 'notas.txt'...

Erro: arquivo 'notas.txt' não encontrado.

Fechando arquivo... (sempre executado)

Caso 3 — erro de leitura

Abrindo 'notas.txt'...

Erro: falha ao ler o arquivo.

Fechando arquivo... (sempre executado)