

Lista de exercicios: Tratamento de Exceção

Gabriel Rodrigues Caldas de Aquino

October 8, 2025

1 Conversão de entrada para número inteiro

Descrição: Solicite ao usuário que digite um número inteiro. Utilize *try/except* para capturar o erro caso o usuário insira um valor que não possa ser convertido para inteiro (por exemplo, letras ou símbolos).

2 Calcular Divisão

Descrição: Peça ao usuário dois números e exiba o resultado da divisão do primeiro pelo segundo. Utilize *try/except* para tratar a exceção de divisão por zero

3 Acesso a elementos de lista

Descrição: Crie uma lista com alguns elementos e solicite ao usuário que digite um índice para acessar um elemento da lista. Utilize *try/except* para tratar o erro caso o índice informado seja inválido (fora do intervalo da lista).

4 Soma de elementos de uma lista

Descrição: Solicite ao usuário que insira uma lista de números separados por espaço. Converta os valores para inteiros e calcule a soma. Utilize *try/except* para tratar possíveis erros durante a conversão.

5 Leitura de arquivo

Descrição: Peça ao usuário o nome de um arquivo para leitura. Utilize *try/except* para tratar a exceção caso o arquivo não exista.

6 Cálculo de média com tratamento de exceções

Descrição: Solicite ao usuário que insira uma lista de números separados por vírgula. Converta os valores para números e calcule a média. Utilize *try/except* para tratar: (i) possíveis erros durante a conversão; e (ii) divisão por zero.

7 Leitura de arquivo com tratamento de múltiplas exceções

Descrição: Peça ao usuário o nome de um arquivo e tente abri-lo para leitura. Utilize *try/except* para tratar as seguintes exceções: (i) caso o arquivo não existir; (ii) Adicione também um tratamento para qualquer outra exceção genérica.

8 Validação de entrada com exceções específicas

Descrição: Solicite ao usuário que insira um número inteiro positivo. Utilize *try/except* para capturar entradas inválidas e levante uma exceção personalizada (use o *raise* com *ValueError*) caso o número seja negativo ou zero.

9 Manipulação de dicionários com tratamento de exceções

Descrição: Crie um dicionário com alguns pares chave-valor. Solicite ao usuário que insira uma chave e exiba o valor correspondente. Utilize *try/except* para tratar o caso em que a chave não existe no dicionário.

10 Uso de *else* e *finally* em blocos *try/except*

Descrição: Crie um programa que solicite ao usuário dois números e exiba o resultado da divisão do primeiro pelo segundo. Utilize *try/except* para capturar possíveis exceções, *else* para exibir uma mensagem quando a operação for bem-sucedida e *finally* para exibir uma mensagem final independente do resultado.

11 BONUS: Faça o tratamento dos possíveis erros que podem acontecer neste código mostrado em aula

```
def calcular_media(valores):
    soma = sum(valores)
    media = soma / len(valores)
    return media

def ler_arquivo(nome_arquivo):
    with open(nome_arquivo, 'r') as file:
        dados = file.read().split(',')
    return [float(dado) for dado in dados]

print("=== Calculadora de Média ===")
valores = input("Digite números separados por vírgula: ").split(',')

valores = [int(x) for x in valores]

print(f"Valores de entrada: {valores}")

media = calcular_media(valores)
print(f"Média do teclado: {media}")

dados_arquivo = ler_arquivo('dados.txt')
print(f"Média do arquivo: {calcular_media(dados_arquivo)}")
```