

Exercícios:

1. **Tratamento de exceções básicas:** Escreva um programa que peça ao usuário dois números e faça a divisão do primeiro pelo segundo. Se o usuário inserir um valor inválido ou tentar dividir por zero, o programa deve exibir uma mensagem de erro apropriada.
2. **Capturando exceções múltiplas:** Crie um programa que peça ao usuário o nome de uma cor e mostre seu valor em RGB de acordo com um dicionário pré-definido. O programa deve tratar exceções caso o nome da cor não exista no dicionário.

```
cores = {'vermelho': (255, 0, 0), 'verde': (0, 255, 0), 'azul': (0, 0, 255)}
```

3. **Bloco `else` e `finally`:** Escreva um programa que solicite um número ao usuário. Se o número for maior que 10, exiba uma mensagem dizendo que o número é válido. Utilize o bloco `else` para imprimir que o programa foi executado com sucesso, e o bloco `finally` para imprimir "Programa encerrado".
4. **Exceções personalizadas:** Escreva uma função que verifica se uma senha possui no mínimo 8 caracteres e pelo menos um número. Se a senha não atender aos requisitos, levante uma exceção com uma mensagem personalizada. Trate a exceção e mostre a mensagem ao usuário.
5. **Simulação de transações:** Crie um programa que simule uma transferência bancária. Peça ao usuário o saldo da conta e o valor da transferência. Caso o saldo seja insuficiente, levante uma exceção do tipo `ValueError` com a mensagem "Saldo insuficiente". Trate a exceção adequadamente e informe o usuário.