

# Fundamentos de Python

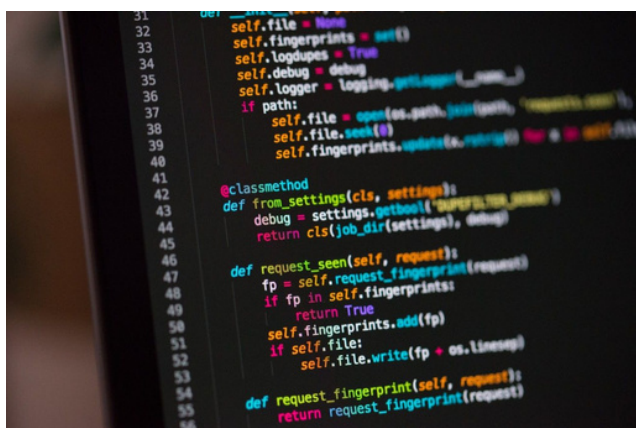
## Tratamento de Erros e Testes

### Objetivo:

Fazer seu código mais seguro e confiável.

#### ⚠ Utilidades do Tratamento de Erros e Testes

- Evita falhas no programa, tratando problemas como divisões por zero ou entradas inválidas.
- Melhora a experiência do usuário, mostrando mensagens claras em vez de travar o sistema.
- Facilita a manutenção, permitindo identificar e corrigir erros mais facilmente.
- Exceções personalizadas ajudam a lidar com situações específicas do seu sistema.
- Bloco finally garante que ações importantes (como fechar arquivos) sejam sempre executadas.



### Blocos try, except, finally

```
try:
    numero = int(input("Digite um número:
"))
    resultado = 10 / numero
    print("Resultado:", resultado)
except ZeroDivisionError:
    print("Erro: divisão por zero!")
except ValueError:
    print("Erro: valor inválido!")
finally:
    print("Operação finalizada.")
```

### Criar Exceções Personalizadas

```
class ErroPersonalizado(Exception):
    pass

def verificar_idade(idade):
    if idade < 18:
        raise ErroPersonalizado("Menor
de idade não permitido!")
    return True

try:
    verificar_idade(16)
except ErroPersonalizado as e:
    print("Erro:", e)
```