

# Programação Funcional

## Projeto de funções

Marco A L Barbosa

[malbarbo.pro.br](http://malbarbo.pro.br)

### Começando

- 1) Quais são as etapas do processo de projeto de funções?
- 2) Qual é o propósito da análise?
- 3) Qual é o propósito da definição dos tipos de dados?
- 4) Quais são as partes que compõem a especificação de uma função?
- 5) Qual é a principal propriedade que uma especificação deve ter para ser considerada adequada?
- 6) O que é a assinatura de uma função?
- 7) Qual é o objetivo inicial dos exemplos no projeto de uma função? E os demais objetivos?
- 8) Se não forem encontrados erros na verificação para os exemplos da especificação, é possível afirmar que a função está isenta de erros? Explique.
- 9) A implementação é a fase mais importante do projeto de funções, verdadeiro ou falso? Explique.
- 10) Como proceder quando um teste falha?
- 11) Qual é o objetivo da revisão?
- 12) Qual é a diferença do resultado da análise e a descrição do propósito da função?

### Praticando

- 13) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
/// Produz True se uma pessoa com *idade* é isento da  
/// tarifa de transporte público, isto é, tem menos  
/// que 18 anos ou 65 ou mais. Produz False caso contrário.  
pub fn isento_tarifa(idade: Int) -> Bool {  
    False  
}  
  
pub fn isento_tarifa_examples() {  
    check.eq(isento_tarifa(17), True)  
    check.eq(isento_tarifa(18), True)  
    check.eq(isento_tarifa(50), False)  
    check.eq(isento_tarifa(65), True)  
    check.eq(isento_tarifa(70), True)  
}
```

- 14) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
/// Conta a quantidade de dígitos de *n*.  
/// Se *n* é 0, então devolve zero.  
/// Se *n* é menor que zero, então devolve a quantidade
```

```

/// de dígitos do valor absoluto de *n*.
pub fn quantidade_digitos(n: Int) -> Int {
    0
}

pub fn quantidade_digitos_examples() {
    check.eq(quantidade_digitos(123), 3)
    check.eq(quantidade_digitos(0), 1)
    check.eq(quantidade_digitos(-1519), 4)
}

```

- 15) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```

/// Produz True se uma pessoa com a *idade* é supercentenária,
/// isto é, tem 110 anos ou mais, False caso contrário.
pub fn supercentenario(idade: Int) -> Bool {
    False
}

pub fn supercentenario_examples() {
    check.eq(supercentenario(101), False)
    check.eq(supercentenario(110), False)
    check.eq(supercentenario(112), True)
}

```

- 16) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```

/// Transforma a string *data* que está no formato "dia/mes/ano"
/// para o formato "ano/mes/dia".
///
/// Requer que o dia e o mês tenham dois dígitos e que
/// o ano tenha quatro dígitos.
pub fn dma_para_amd(data: String) -> String {
    data
}

pub fn dma_para_amd_examples() {
    check.eq(dma_para_amd("19/07/2023"), "2023/07/19")
    check.eq(dma_para_amd("01/01/1980"), "1980/01/01")
    check.eq(dma_para_amd("02/02/2002"), "2002/02/20")
}

```

- 17) Escreva a especificação para a seguinte implementação de função. Avalie se a sua especificação está boa, verificando se ela sozinha é suficiente para um desenvolvedor fazer uma implementação da função.

```

pub fn aumenta(valor: Float, porcentagem: Float) -> Float {
    valor *. (1.0 +. porcentagem /. 100.0)
}

```

- 18) Escreva a especificação para a seguinte implementação de função. Avalie se a sua especificação está boa, verificando se ela sozinha é suficiente para um desenvolvedor fazer uma implementação da função.

```

pub fn tamanho_nome(nome: String) -> Int {
    case string.length(nome) <= 4 {
        True -> "curto"
        False -> case string.length(nome) <= 10 {
            True -> "médio"
            False -> "longo"
        }
    }
}

```

}  
}

- 19) Projete uma função que adicione um ponto final a uma frase se ela não acabar com um.
- 20) Projete uma função que determine se uma palavra tem um traço ("-") no meio, como por exemplo, "l~~e~~ro-l~~e~~ro". Não use nenhuma condicional na implementação.
- 21) Projete uma função que encontre o máximo entre três números dados.
- 22) Projete uma função que receba como parâmetro uma string e um número natural  $n$  e substitua os primeiros  $n$  caracteres da string por  $n$  letras  $x$ .

## Resolvendo problemas

- 23) Você está fazendo um programa e precisa verificar se um texto digitado pelo usuário está de acordo com algumas regras. A regra “sem espaços extras” requer que o texto não comece e não termine com espaços. Projete uma função que verifique se um texto qualquer está de acordo com a regra “sem espaços extras”.
- 24) Cada cidadão de um país, cuja moeda chama dinheiro, tem que pagar imposto sobre a sua renda. Cidadãos que recebem até 1000 dinheiros pagam 5% de imposto. Cidadãos que recebem entre 1000 e 5000 dinheiros pagam 5% de imposto sobre 1000 dinheiros e 10% sobre o que passar de 1000. Cidadãos que recebem mais do 5000 dinheiros pagam 5% de imposto sobre 1000 dinheiros, 10% de imposto sobre 4000 dinheiros e 20% sobre o que passar de 5000. Projete uma função que calcule o imposto que um cidadão deve pagar dado a sua renda.
- 25) Uma palavra duplicada é formada pela ocorrência de duas partes iguais, separadas ou não por hífen. Por exemplo, as palavras xixi, mimi, lero-lero e mata-mata são palavras duplicadas. Projete uma função que verifique se uma palavra é duplicada.
- 26) Um construtor precisa calcular a quantidade de azulejos necessários pra azulejar uma determinada parede. Cada azulejo é quadrado e tem 20cm de lado. Ajude o construtor e defina uma função que receba como entrada o comprimento e a altura em metros de uma parede e calcule a quantidade de azulejos inteiros necessários para azulejar a parede. Considere que o construtor nunca perde um azulejo e que recortes de azulejos não são reaproveitados.
- 27) Rotacionar uma string  $n$  posições a direita significa mover os últimos  $n$  caracteres da string para as primeiras  $n$  posições da string. Por exemplo, rotacionar a string "marcelio" 5 posições a direita produz a string "celiomar". Projete uma função que receba como entrada uma string e um número  $n$  e produza uma nova string rotacionando a string de entrada  $n$  posições a direita.
- 28) No período de 2015 à 2016 todos os números de telefones celulares no Brasil passaram a ter nove dígitos. Na época, os números de telefones que tinham apenas oito dígitos foram alterados adicionando-se o 9 na frete do número. Embora oficialmente todos os número de celulares tenham nove dígitos, na agenda de muitas pessoas ainda é comum encontrar números registrados com apenas oito dígitos. Projete uma função que adicione o nono dígito em um dado número de telefone celular caso ele ainda não tenha o nono dígito. Considere que os números de entrada são dados com o DDD entre parênteses e com um hífen separando os últimos quatro dígitos. Exemplos de entradas: (44) 9787-1241, (51) 95872-9989, (41) 8876-1562. A saída deve ter o mesmo formato, mas garantindo que o número do telefone tenha 9 dígitos.
- 29) Muitos letrados exibem mensagens que tem mais caracteres do que eles podem exibir, para isso, eles exibem apenas uma porção da mensagem que é alterada com o passar do tempo. Por exemplo, em um letrado de 20 caracteres, a mensagem "Promocao de sorvetes, pague 2 leve 3!" é exibida como Promocao de sorvetes no momento 0, como "romocao de sorvetes," no momento 1, "omocao de sorvetes, ", no momento 3, e assim por diante até que no momento 17 é exibido "tes, pague 2 leve 3!". O momento sempre aumenta, e após chegar no final da mensagem ela começa a ser exibida novamente, nesse caso, no momento 18 é exibido "es, pague 2 leve 3! " e no momento 19 é exibido "s, pague 2 leve 3! P", onde o P é o início da mensagem. Projete uma função que determine os

caracteres de uma mensagem que devem ser exibidos em um determinado momento em um letreiro que pode exibir um determinado número de caracteres. Assuma que o número de caracteres da mensagem é maior do que o do letreiro.

- 30) (Desafio) Um número inteiro positivo é palíndromo se quando lido da direita para a esquerda ou da esquerda para a direita é idêntico. Ex: 9119, 1221, 5665, 7337. Projete uma função que verifique se um dado número inteiro de 4 dígitos é palíndromo, considere que o valor de entrada é o próprio número e não os quatro dígitos que compõem o número. É possível modificar a sua função de maneira que ela funcione para qualquer número de entrada e não apenas para números de 4 dígitos?