

```
;;=====;;  
;; How to Design Programs, 2ª edição  
;; https://htdp.org  
;;-----;;  
;; Capítulos: (diversos)  
;;-----;;  
;; Resumo por: Abrantes Araújo Silva Filho  
;; abrantesasf@gmail.com  
;;=====;;
```

RECEITA DE DESIGN DE FUNÇÃO (RDF)

1) DEFINIÇÃO DE DADOS:

Analise o problema, tipicamente escrito em palavras, e:

- a) Identifique a INFORMAÇÃO que deve ser representada, e defina como ela será representada pelos DADOS da linguagem de programação escolhida. Formule as DEFINIÇÕES DE DADOS. Ex.:
 - ; Representação: a temperatura será um número
 - ; Interpretação: em graus Celsius

2) ASSINATURA, DECLARAÇÃO DE PROPÓSITO E CABEÇALHO DA FUNÇÃO

Para cada função a ser desenvolvida, prepare:

- a) ASSINATURA DA FUNÇÃO: é um comentário que indica quantos inputs a função necessita, quais as classes dos inputs, e o que a função produzirá. Ex.:
 - ; Número String Imagem -> Imagem
- b) DECLARAÇÃO DE PROPÓSITO: é um comentário que resume o propósito da função, em uma ou poucas linhas. Deve responder claramente à pergunta: "O que essa função calculará?". Em programas com múltiplas funções, uma declaração de propósito geral do programa deve ser fornecida no começo do arquivo. Ex.:
 - ; adiciona s à img, y pixels do topo e 10 da esquerda
- c) CABEÇALHO DA FUNÇÃO: é um "mockup" da função que será construída, mas já com nome definido e parâmetros (com nomes significativos) já definidos. O corpo da função pode retornar apenas um único valor da mesma classe do output desejado, pois será construído posteriormente. Ex.:
 - (define (add-image y s img)
 - (empty-scene 100 100))

3) EXEMPLOS FUNCIONAIS:

Ilustre a assinatura e a declaração de propósito. Prepare:

- a) EXEMPLOS FUNCIONAIS: construa alguns exemplos funcionais indicando os inputs e os outputs esperados. Escolha exemplos representativos da funcionalidade esperada da função.

4) FAÇA O INVENTÁRIO E CONSTRUA UM TEMPLATE:

Faça um inventário das coisas que são necessárias para construir a função, para entender claramente quais são os inputs e o que a função precisa calcular.

Prepare um template da função:

- a) TEMPLATE: substitua o corpo da função no cabeçalho criado e introduza os parâmetros no corpo da função, com "..." para indicar que não é uma função completa, é apenas um template que precisa ser completado.

5) CODIFIQUE A FUNÇÃO:

Agora é a hora de codificar a função, ou seja, programar a funcionalidade esperada, substituindo o template por código real.

6) TESTE A FUNÇÃO:

Teste a função com os exemplos funcionais definidos e novos casos de teste.

- a) Testes com exemplos funcionais: garantem que, no mínimo, a função produz o resultado esperado com os inputs mais básicos.
 - b) Testes extensivos: planeje casos de teste que irão desafiar a função: inputs errados, inputs fora de limites, etc. Garanta que a função produza o output correto em todos os testes extensivos.
-

Orientações Gerais Complementares à RDF

- a) Defina 1 função para cada tarefa
- b) Elimine os números mágicos (Um programa, Várias definições): para cada constante do problema, introduza uma definição de constante.
- c) Para cada função, crie testes unitários.
- d) Estabeleça pontos únicos de controle para todos os aspectos das funções/programas.