Programação Funcional

Prioriza esse formalismo matemático na formulação das soluções para os problemas computacionais propostos. A idéia é pensar no **algoritmo** como sendo uma grande e única função formada por uma **combinação** de várias outras pequenas **funções**, cada uma resolvendo uma pequena parte do problema.

Organizando o pensamento computacional

Uma boa prática para elaboração do algoritmo de solução de um problema e posterior codificação desse algoritmo em uma linguagem de programação é:

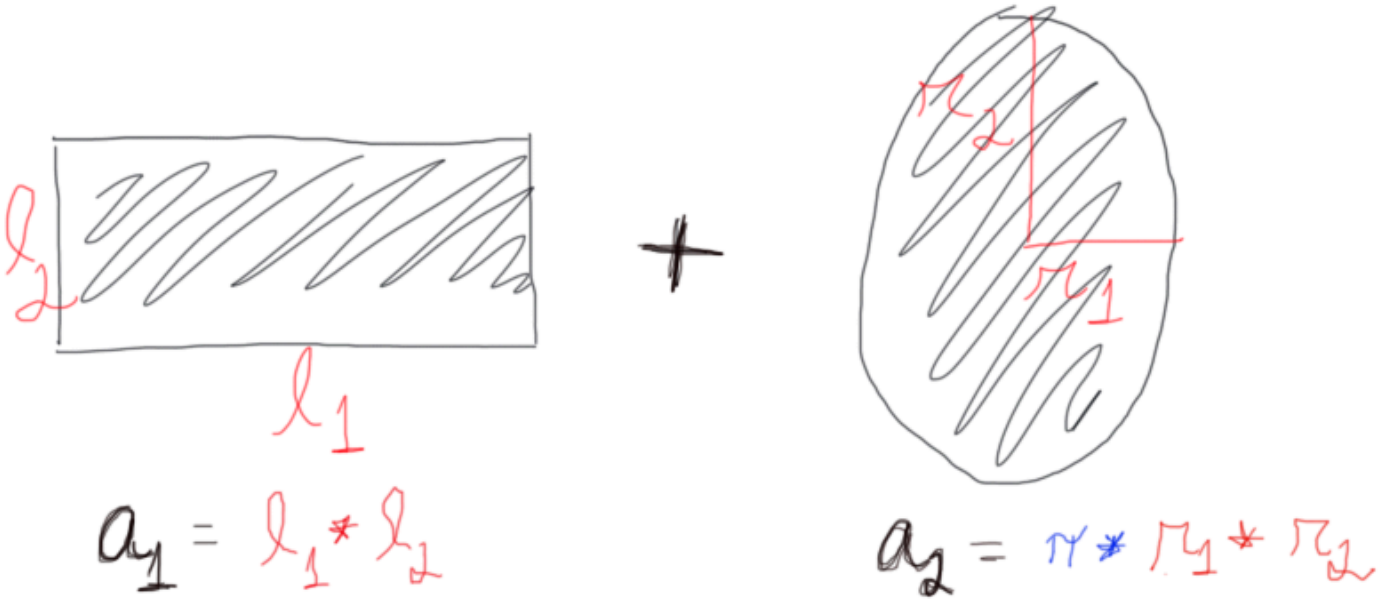
1. identificar o problema principal a ser tratado,
2. identificar sub-problemas dos quais aquele depende,
3. prover algoritmos de solução para todos,
4. codificar essas soluções na linguagem desejada

Ao seguir os passos 1, 2 e 3, estaremos redigindo a função principal no topo das anotações e seguindo para "baixo" escrevendo as sub-funções necessárias.

Entretanto, ao codificarmos de fato essas funções, precisamos inverter essa ordem, uma vez que o compilador precisa conhecer de antemão todas as funções que por ventura sejam demandadas por outras mais à frente no código.

Por isso, faz sentido considerar que elaboramos o algoritmo "DE CIMA PARA BAIXO" e o código "DE BAIXO PARA CIMA".

> **[EXEMPLO] Crie um programa que calcule a soma das áreas de duas figuras geométricas distintas: um retângulo e uma elipse.**
>
> ***Algoritmo visual***



Algoritmo em pseudocódigo

"DE CIMA PARA BAIXO"

resultado $\rightarrow fsoma(a1, a2)$

a1 $\rightarrow fareaRet(l1, l2)$
a2 $\rightarrow fareaEli(r1, r2)$

Subproblemas são gerados...
$l1 =?, l2 =?, r1 =?, r2 =?$
$fsoma(a1, a2) =?$
$fareaRet(l1, l2) =?$
$fareaEli(r1, r2) =?$

... e resolvidos:
$l1, l2, r1, r2$ são pré-deifinidos ou fornecidos pelo usuário
$fsoma(x, y) = x + y$
$fareaRet(x, y) = x * y$
$fareaEli(x, y) = \pi * x * y$

Programa em Javascript

"DE BAIXO PARA CIMA"

```
...
resultado = fsoma(a1,a2)
```

```
...
const a1 = fareaRet(l1,l2)
const a2 = fareaEli(r1,r2)

resultado = fsoma(a1,a2)
```

```
...
const l1 = 6.1 //ou fornecido pelo usuário
const l2 = 4.4 //ou fornecido pelo usuário
const r1 = 3.0 //ou fornecido pelo usuário
const r2 = 5.3 //ou fornecido pelo usuário

const a1 = fareaRet(l1,l2)
const a2 = fareaEli(r1,r2)

resultado = fsoma(a1,a2)
```

```
function fareaEli(x, y) {
    const pi = 3.1415
    return pi*x*y
}

function fareaRet(x, y) {
    return x*y
}

function fsoma(x, y) {
    return x + y
}

const l1 = 6.1 //ou fornecido pelo usuário
const l2 = 4.4 //ou fornecido pelo usuário
const r1 = 3.0 //ou fornecido pelo usuário
const r2 = 5.3 //ou fornecido pelo usuário

const a1 = fareaRet(l1,l2)
const a2 = fareaEli(r1,r2)

resultado = fsoma(a1,a2)

console.log(resultado)
```

Notação de funções em Javascript