Recursão

Prof. Dr. Eleandro Maschio
Tecnologia em Sistemas para Internet
Campus Guarapuava
Universidade Tecnológica Federal do Paraná (UTFPR)

Recursão

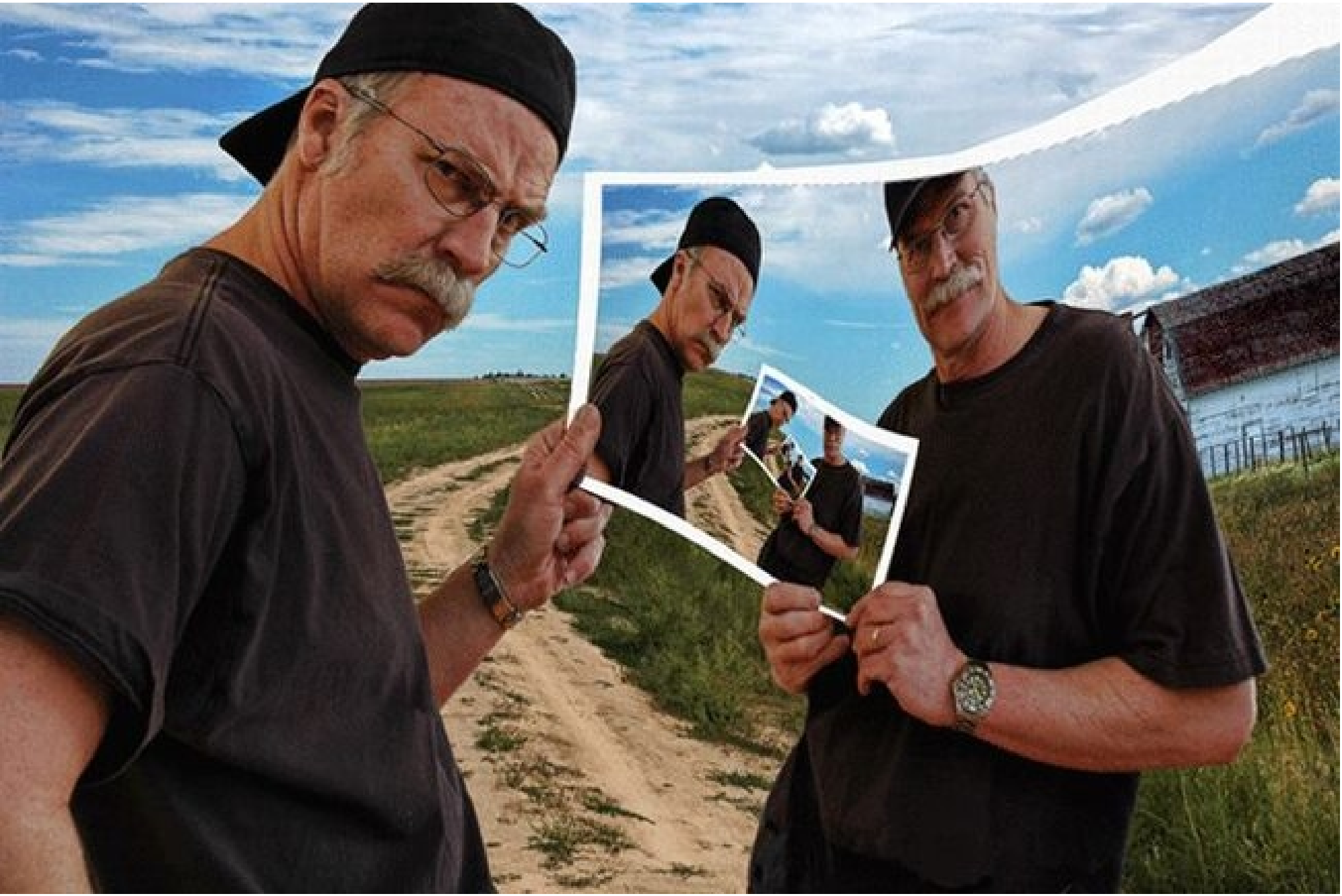
As boas notícias:

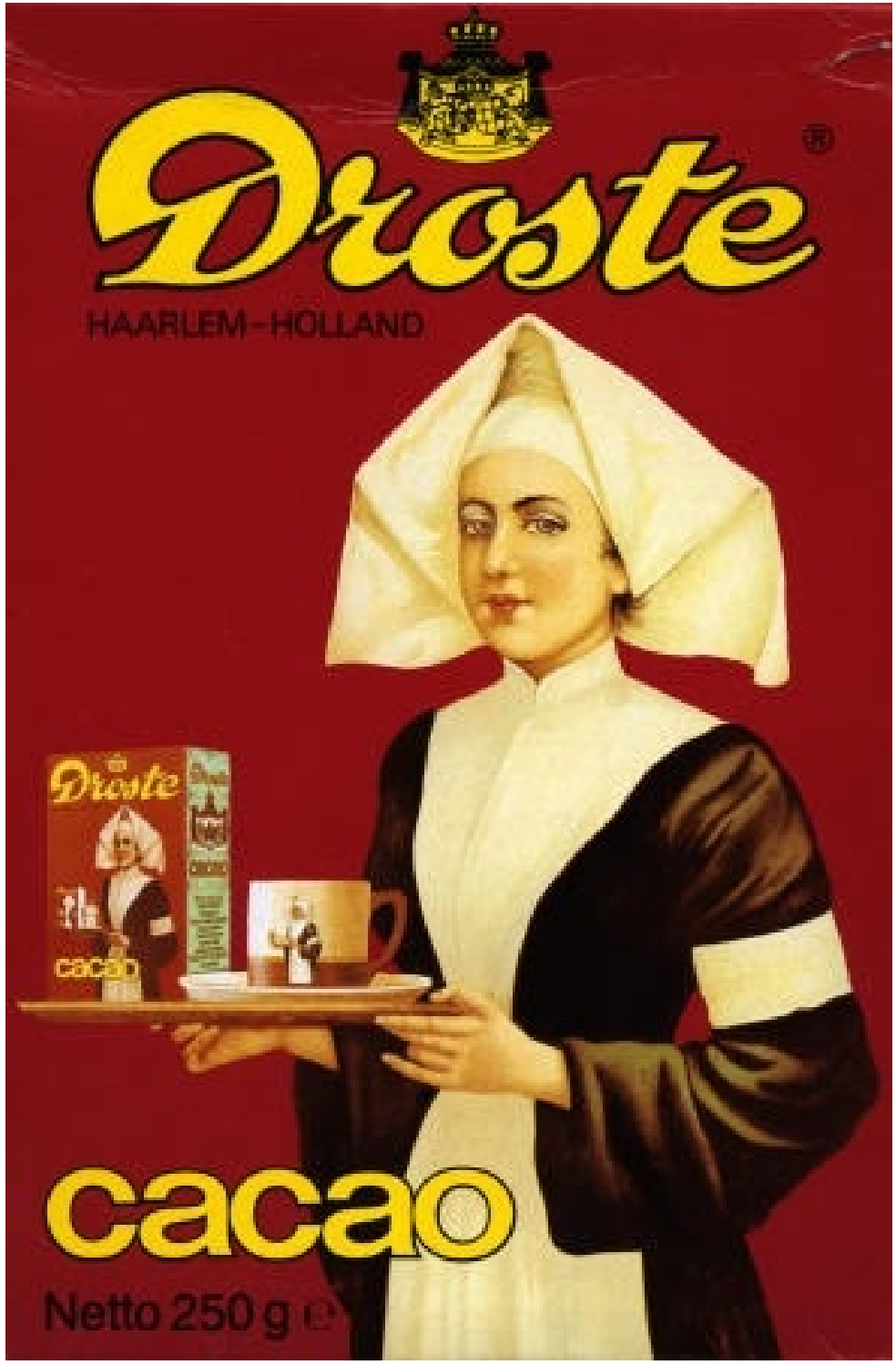
- Muitos daqueles que tem dificuldades com Estruturas de Repetição vêem na Recursividade uma forma mais próxima de expressar seu raciocínio através de uma linguagem de programação. Dentro disto, vale rever quais são estas dificuldades anteriores e transpô-las agora com o raciocínio melhor estruturado.
- Mesmo aqueles que possuem domínio sobre as Estruturas de Repetição aprenderão um recurso poderoso para resolução de problemas.
- Tanto nas disciplinas subseqüentes quanto no mercado de trabalho, ter pleno domínio de Recursão é característica básica de um bom programador.

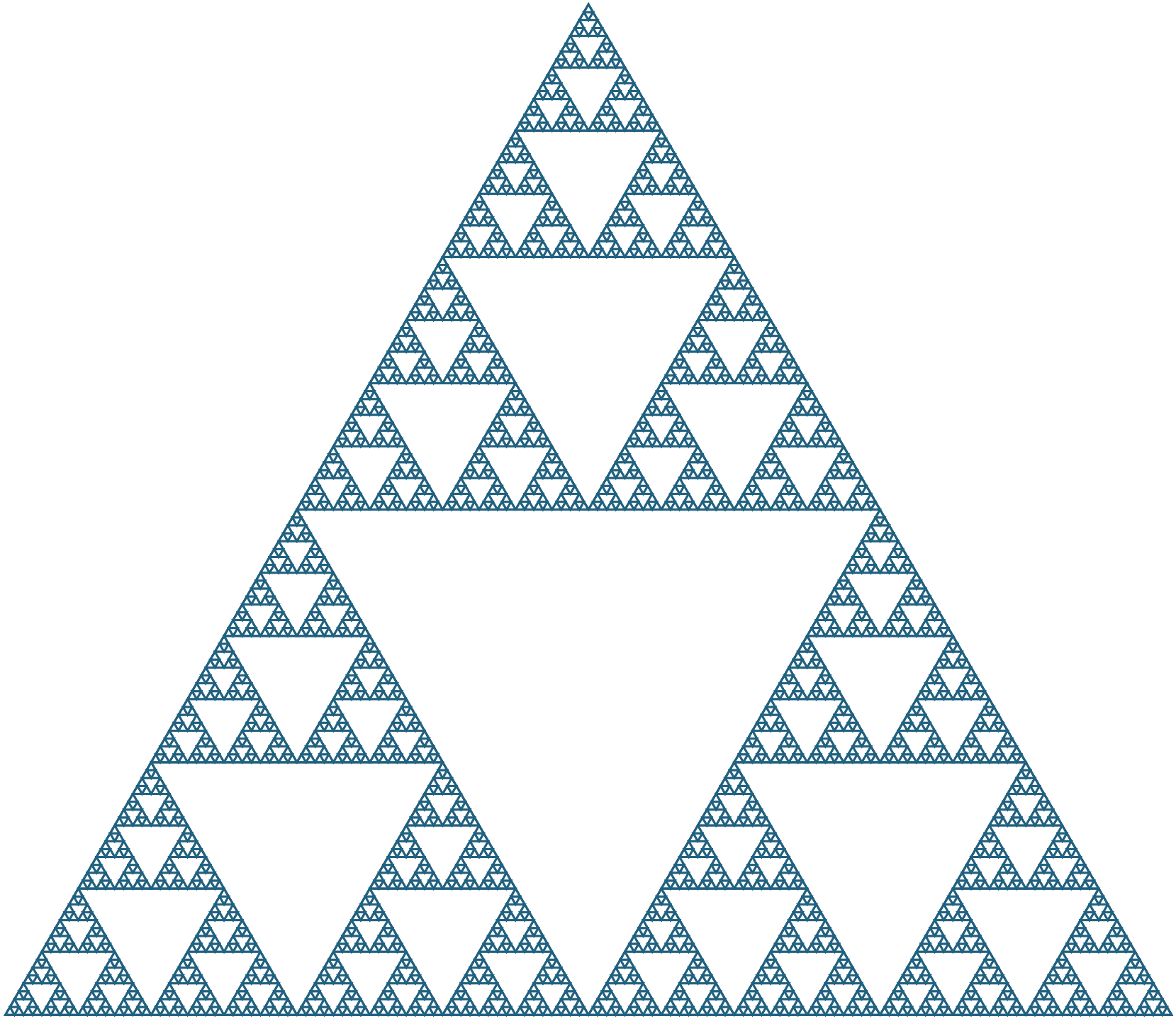
Recursão

Recursão é o processo de definir algo em termos de si mesmo.

Freqüentemente é também chamado de *definição circular*.




Droste
HAARLEM - HOLLAND
Droste
cacao
cacao
Netto 250 g



Recursão

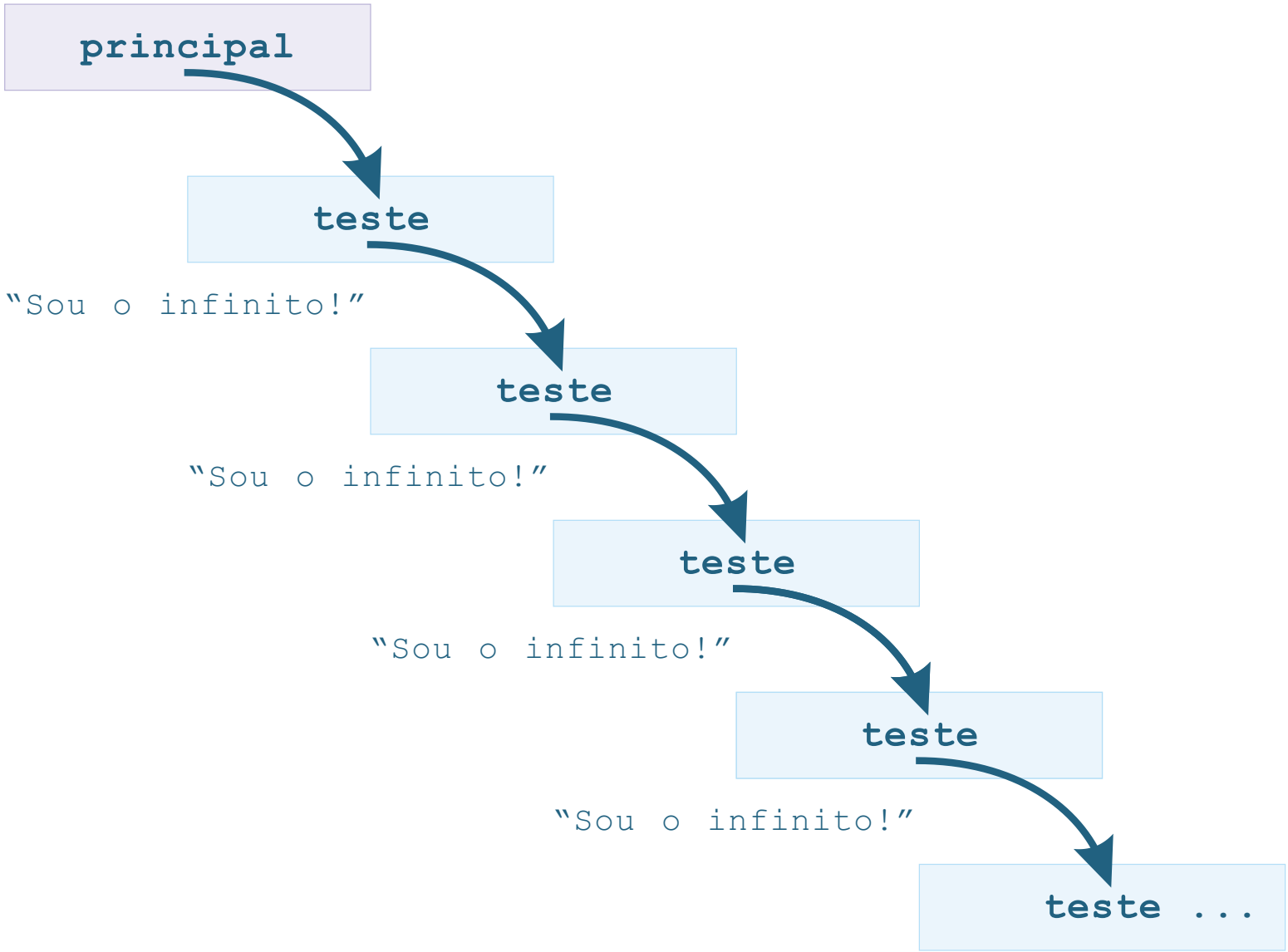
- Uma solução é denominada recursiva se ela é definida em termos de si própria.
- Existem muitos problemas que são definidos intuitivamente de forma recursiva.
- Na Programação de Computadores, uma sub-rotina (procedimento ou função) é recursiva quando em seu corpo há uma chamada para si mesma.

Um Primeiro Exemplo

Experimente este exemplo pouco funcional:

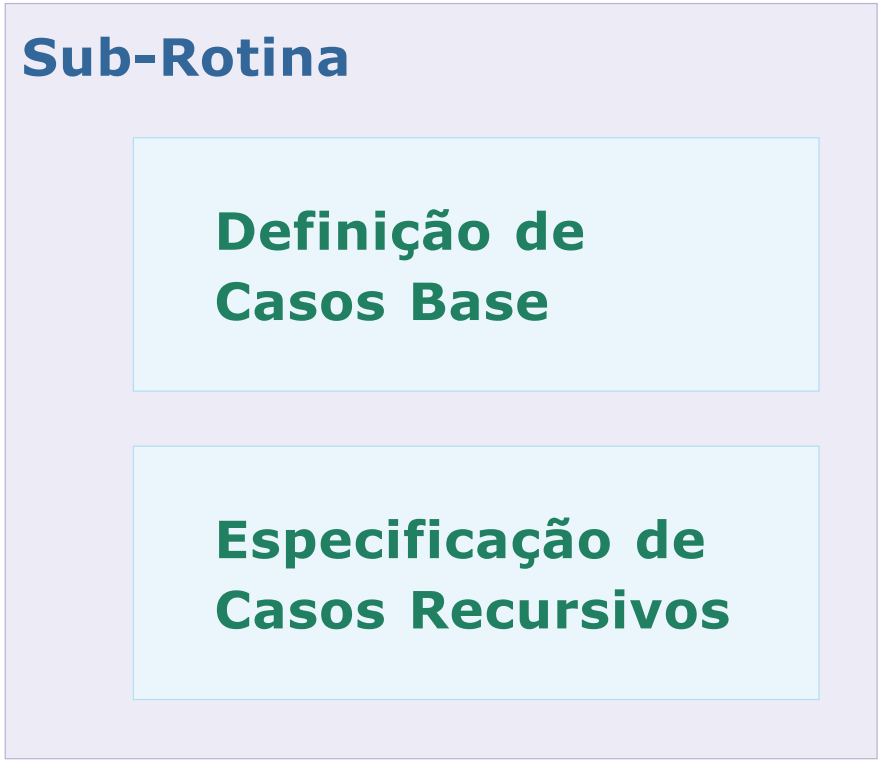
```
int main ()
{
   printf("Prepare-se...\n\n");
   teste();

   return 0;
}

void teste()
{
   printf("Eu sou o infinito!\n");
   teste();
}
```

Um Primeiro Exemplo



Construção de uma Sub-Rotina Recursiva

A recursão deve definir alguns poucos casos base (freqüentemente apenas um) e então especificar regras para formular casos complexos em termos destes casos mais simples.

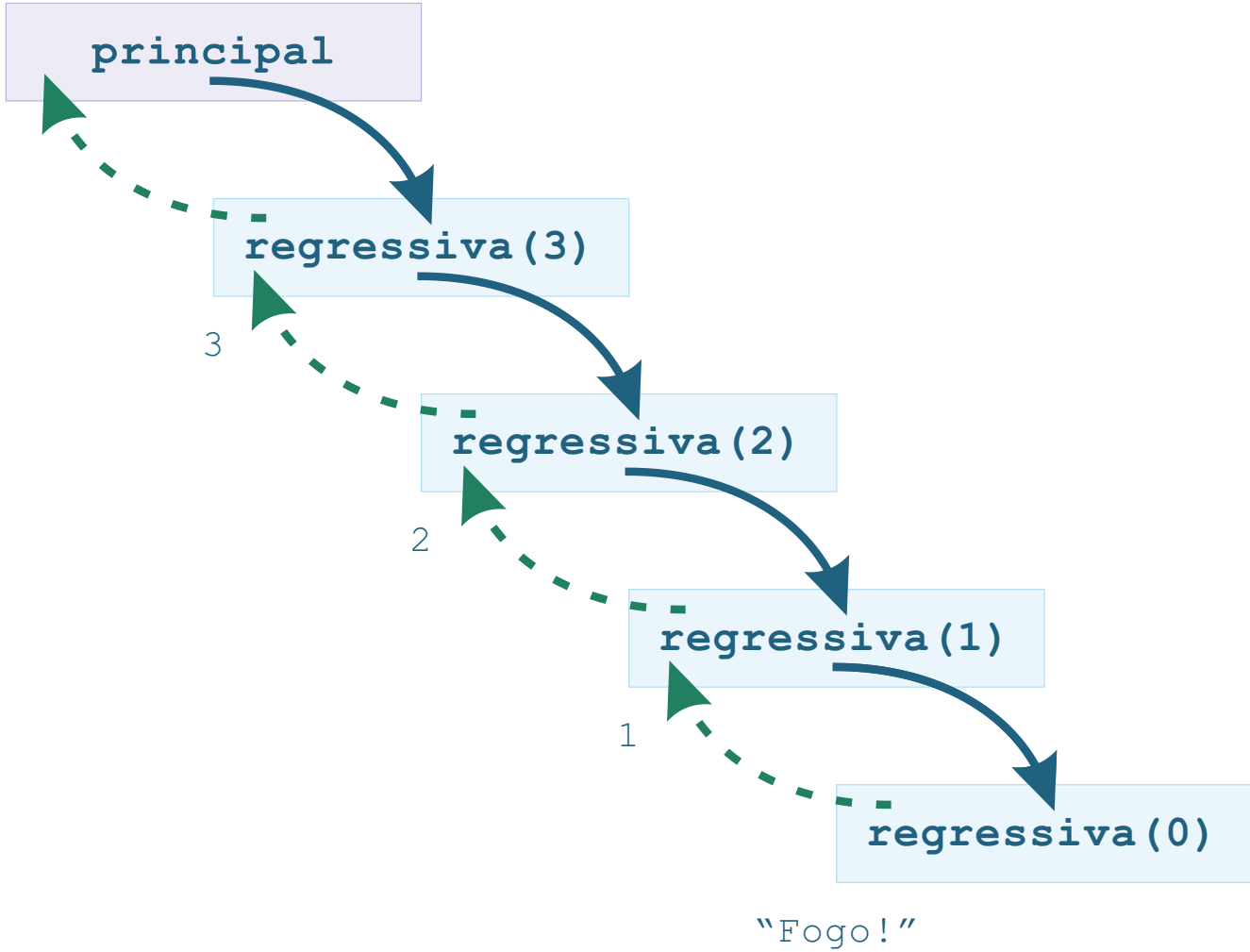


Contagem Regressiva

Realizar a contagem regressiva de ***n*** até zero.

```
int main ()
{
   regressiva(3);
   return 0;
}

void regressiva(int n)
{
   if (n > 0)
   {
      printf("%d\n", n);
      regressiva(n-1);
   }
   else
      printf("Fogo!\n");
}
```

Contagem Regressiva

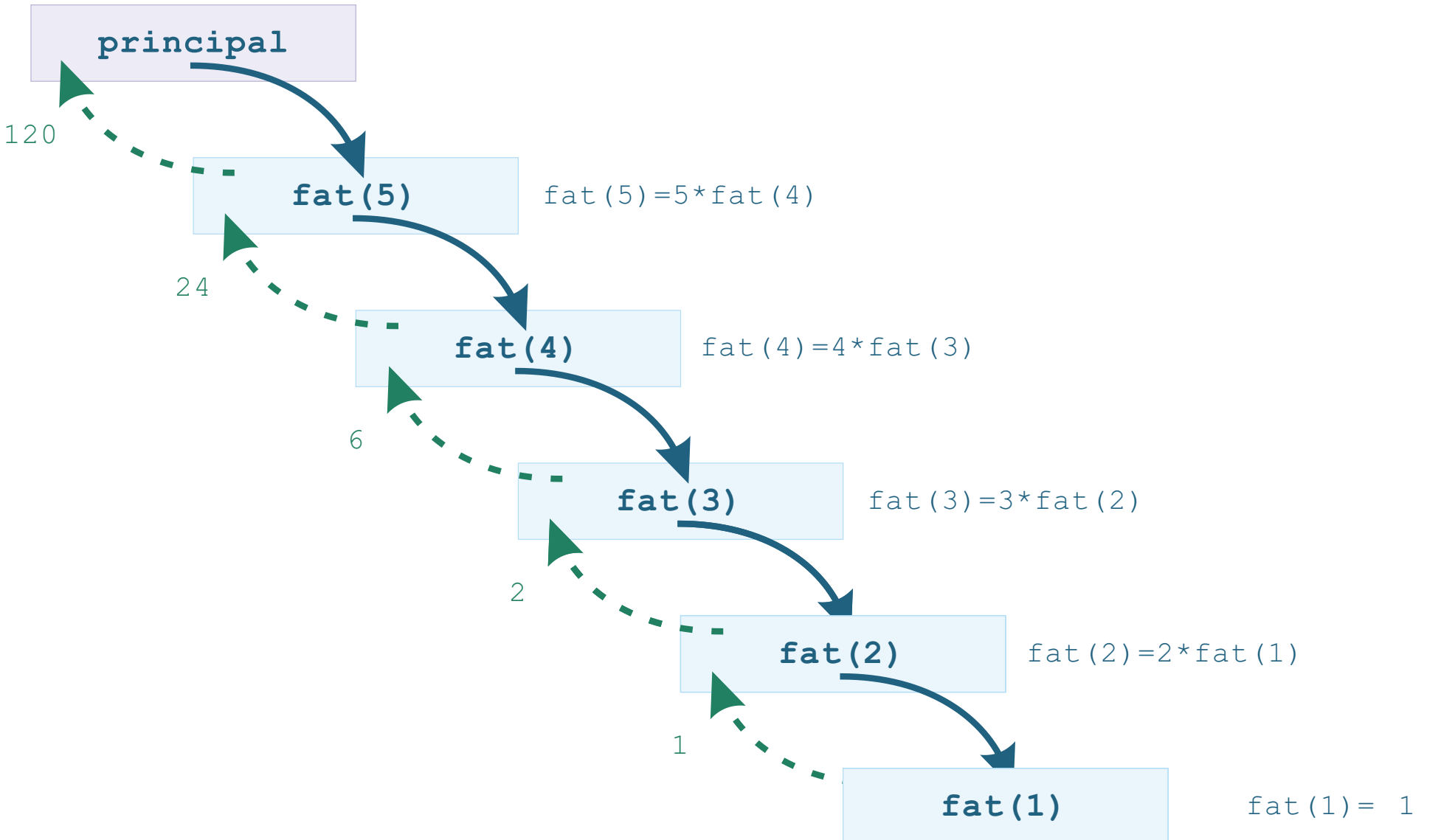


Fatorial

Retornar o fatorial de ***n***, sendo *n* >= 0.

```
int main ()
{
   printf("%d", fat(4));
   return 0;
}

int fat(int n)
{
   if (n <= 1)
      return 1;
   else
      return n * fat(n-1);
}
```

Fatorial


principal
120
fat(5)
fat(5)=5*fat(4)
24
fat(4)
fat(4)=4*fat(3)
6
fat(3)
fat(3)=3*fat(2)
2
fat(2)
fat(2)=2*fat(1)
1
fat(1)
fat(1)= 1

"To understand recursion, we must first understand recursion."

Um Filme:
Inception (A Origem)