



Programação Funcional com Elixir

The background of the slide features a series of overlapping, semi-transparent gray shapes that resemble stylized leaves or petals. These shapes are layered on top of each other, creating a sense of depth and movement. The colors range from light gray to a slightly darker shade, all set against a plain white background.

Recursão

Recursão

- Recursão

- Em linguagens imperativas, a repetição é feita usando características de iteração, tais como loops (while, for), que dependem de um estado mutável.
- Em programação funcional nós temos estados imutáveis, então, nós precisamos de uma abordagem diferente, neste caso, funções recursivas.
- Em resumo... “Uma função recursiva é aquela que chama a si própria”

Recursão

$$n! = \begin{cases} 1, & \text{se } n = 0 \text{ ou } n = 1 \\ n(n-1)!, & \text{se } n \geq 2 \end{cases}$$

```
funcao fatorial(n){  
    se (n==0 ou n==1)  
        retorne 1;  
  
    se (n > 1)  
        retorne n * fatorial(n-1);  
}
```

Recursão

$$n! = \begin{cases} 1, & \text{se } n = 0 \text{ ou } n = 1 \\ n(n-1)!, & \text{se } n \geq 2 \end{cases}$$

```
int fatorial(int n)
{
    if (n >= 2)
        return n * fatorial(n - 1);
    else
        return 1;
}
```

Recursão

- O que podemos observar?
 - **1. Ponto de Parada:** o ponto de parada da recursividade é resolvido sem utilização de recursividade, sendo este ponto geralmente um limite superior ou inferior da regra geral.
 - **2. Regra Geral:** o método geral da recursividade reduz a resolução do problema através da invocação recursiva de casos mais pequenos, sendo estes casos mais pequenos resolvidos através da resolução de casos ainda mais pequenos, e assim sucessivamente, até atingir o ponto de parada que finaliza o método.

Recursão

- Implementando em Elixir

- `defmodule Factorial do`
- `def of(0), do: 1`
- `def of(n) when n > 0 do`
- `n * of(n-1)`
- `end`
- `end`

Recursão

- Assim, aprendemos mais um dos princípios do paradigma funcional.
 - First-Class functions ✓
 - Pure functions
 - Immutable variables ✓
 - ✓ Recursion
 - Nonstrict evaluation
 - Statements ✓
 - Pattern matching ✓