



Programação Funcional com Elixir



Recursão com Funções Anônimas

Recursão com Funções Anônimas

- Até agora usamos funções nomeadas para criar funções recursivas, mas no dia-a-dia vamos encontrar situações onde as funções são anônimas, então, nesses casos, como faremos a recursão?
- Vejamos esse exemplo...

Recursão com Funções Anônimas

- Neste primeiro exemplo, percebemos que normalmente não podemos usar funções anônimas para fazer recursão...

```
factorial = fn
  0 -> 1
  x when x > 0 -> x * factorial.(x - 1)
end
** (CompileError) undefined function factorial/0
```

- Esse erro ocorre devido a função que ainda não existe ser chamada para recursão.

Recursão com Funções Anônimas

- Agora observe o exemplo abaixo, onde usamos uma função anônima envolvendo uma outra função anônima para resolver o nosso problema...

```
fact_gen = fn me ->
  fn
    0 -> 1
    x when x > 0 -> x * me.(me).(x - 1)
  end
end
```

```
iex> factorial = fact_gen.(fact_gen)
iex> factorial.(5)
120
```

```
iex> factorial.(10)
3628800
```

Recursão com Funções Anônimas

- Recapitulando...
 - A função **fact_gen** é uma função anônima que sabe como construir uma função de fatorial.
 - Ela espera que ela mesma seja passada como argumento e isso é o que faz a função anônima produzir a função factorial em memória.

Recursão com Funções Anônimas

- A pegadinha fica por conta de na hora do uso ter que gerar a função antes de usá-la (**factorial = fact_gen(fact_gen)**), justamente para ela poder existir em memória antes do uso.
- No entanto essa abordagem não é nada expressiva e apenas em casos realmente necessários essa técnica é usada.