

07 - Funções anônimas e Funções de alta ordem

André Rauber Du Bois
dubois@inf.ufpel.edu.br
Computação - CDTec - UFPel

Funções anônimas

- São funções definidas dinamicamente no código sem a necessidade de um Módulo
- Funções anônimas são declaradas com a palavra **fn**:

```
iex(1)> soma = fn(x,y) -> x + y end
```

```
#Function<43.65746770/2 in :erl_eval.expr/5>
```

- Criou uma função anônima que soma dois valores e associou ela ao nome **soma**
- **Uso:**

```
iex(2)> soma.(20,100)
```

```
120
```

```
iex(3)>
```

Funções de Alta Ordem

- Funções de alta ordem são funções que recebem funções como argumento
- Funções de alta ordem em Elixir, geralmente recebem como argumento funções anônimas

A função **map**

- A função **map** é uma função de alta ordem muito usada em programação funcional
- A função **map** recebe como argumentos uma função e uma lista e aplica essa função a todos os elementos da lista
- A função **map** já existe pré-definida em Elixir no módulo **Enum**

```
iex(1)> Enum.map([1,2,3,4,5], fn(x) -> x + 1 end)
```

```
[2, 3, 4, 5, 6]
```

```
iex(2)> Enum.map([1,2,3,4,5], fn(x) -> x * x end)
```

```
[1, 4, 9, 16, 25]
```

Como é definida a função **map**?

Como é definida a função **map**?

```
defmodule Aula7 do  
  def map([], _f), do: []  
  def map([h|t], f), do: [f.(h) | map(t, f)]  
end
```

Captura de funções

- Para que uma função definida em um módulo possa ser passada como argumento para uma função de alta ordem, a mesma deve ser **capturada** usando o operador **&**
- A captura transforma a função em um valor

```
defmodule Aula7 do
  def inc(x), do: x + 1
end
```

```
iex(1)> Enum.map([1,2,3,4,5], &Aula7.inc/1)
```

```
[2, 3, 4, 5, 6]
```

- A sintaxe de captura também pode ser usada como uma notação simplificada para definição de funções anônimas:

```
iex(1)> soma = &(&1 + &2)
```

```
&:erlang.+ /2
```

```
iex(2)> soma.(8,12)
```

```
20
```

```
iex(3)> Enum.map([1,2,3,4], &(1 + &1))
```

```
[2, 3, 4, 5]
```

- Aqui **&1** significa o primeiro argumento da função, **&2** o segundo, etc...