



Programação Funcional com Elixir



Cláusulas de Guarda

Cláusulas de Guarda

- Imagine as funções abaixo que servem para somar até o número informado...

- `defmodule Sum do`
- `def to(1), do: 1`
- `def to(n), do: n + to(n-1)`
- `end`
-
- **`Sum.to(5)`**

Cláusulas de Guarda

- Agora imagine se vc passar um número negativo, -1 por exemplo...
- Concorda que isso vai virar um loop infinito?

```
○ defmodule Sum do
○   def to(1), do: 1
○   def to(n), do: n + to(n-1)
○ end
```

Cláusulas de Guarda

- Como evitar que isso aconteça?
- Uma das formas é usando Cláusulas de Guarda (Guard Clauses)

```
o defmodule Sum do
o   def to(1), do: 1
o   def to(n) when n > 0, do: n + to(n-1)
o end
```

Cláusulas de Guarda

- Também é possível usar múltiplas cláusulas, veja:

- `defmodule Sum do`
- `def to(1), do: 1`
- `def to(n) when n > 0 and is_integer(n), do: n + to(n-1)`
- `end`

Cláusulas de Guarda

- Para conhecer um pouco mais sobre Cláusulas de Guarda, e quais expressões podem ser usadas, consulte a documentação oficial.
 - `https://hexdocs.pm/elixir/master/guards.html`