



Programação Funcional com Elixir



Binários, Strings e Charlists

Binários, Strings e Charlists

- Strings são binários codificados em UTF-8, por isso a quantidade de bytes para representar uma string pode variar.
 - `byte_size("Elixir é legal!")`
 - `String.length("Elixir é legal!")`

Binários, Strings e Charlists

- Para saber qual o codepoint do caractere, use ? antes do mesmo.
 - ?a
 - ?é

Codepoints são caracteres Unicode que podem ser representados por **um ou mais bytes**.

Binários, Strings e Charlists

- Para conhecer a representação binária de uma String, concatene ela com `<<0>>`
 - `"Elixir é legal!" <> <<0>>`

Binários, Strings e Charlists

- Para conhecer uma string a partir dos binários, basta atribuí-la a uma variável.
 - `abc = <<69, 108, 105, 120, 105, 114, 32, 195, 169, 32, 108, 101, 103, 97, 108, 33>>`

Binários, Strings e Charlists

- Codepoints são caracteres Unicode que podem ser representados por **um ou mais bytes**.
 - ?é
 - "é" <> <<0>>

Binários, Strings e Charlists

- Internamente, o Elixir representa as Strings como uma sequência de Bytes ao invés de um array/vetor de caracteres, no entanto, caso seja necessário trabalhar com arrays de caracteres, podemos usar as **Charlists** bastando para isso usar aspas simples ao invés de duplas, veja:

- `'Elixir é legal!'`
- `"Elixir é legal!" <> <<0>>`
- `to_charlist("Elixir é legal!")`

Binários, Strings e Charlists

- Conheça mais sobre binários, strings e charlists na documentação oficial...
 - `https://elixir-lang.org/getting-started/binaries-strings-and-char-lists.html`