

Curso Elixir

Sección 5

Pattern Matching

- The match operator

Anteriormente usamos el operador = para asignar valores, en Elixir el operador = es realmente el operador de coincidencia

```
iex> x = 1
1
iex> 1 = x
1
iex> 2 = x
** (MatchError) no match of right hand side value: 1
```

- Pattern matching

El operador de coincidencia no solo se utiliza para comparar valores simples, sino que también es útil para desestructurar tipos de datos más complejos. Por ejemplo, podemos combinar patrones en tuplas:

```
iex> {a, b, c} = {:hello, "world", 42}
{:hello, "world", 42}
iex> a
:hello
iex> b
"world"
```

Se producirá un error de coincidencia de patrón si los lados no pueden coincidir

```
iex> {a, b, c} = {:hello, "world"}
** (MatchError) no match of right hand side value: {:hello, "world"}
iex> {a, b, c} = [:hello, "world", 42]
** (MatchError) no match of right hand side value: [:hello, "world", 42]
```

Más interesante aún, podemos coincidir en valores específicos. El siguiente ejemplo afirma que el lado izquierdo solo coincidirá con el lado derecho cuando el lado derecho es una tupla que comienza con el átomo :ok

```
iex> {:ok, result} = {:ok, 13}
{:ok, 13}
iex> result
13

iex> {:ok, result} = {:error, :oops}
** (MatchError) no match of right hand side value: {:error, :oops}
```

Podemos usar el patrón con listas

```
iex> [a, b, c] = [1, 2, 3]
[1, 2, 3]
```

```
iex> a
1
```

Una lista también admite coincidencias en su propia cabeza y cola:

```
iex> [head | tail] = [1, 2, 3]
iex> [1, 2, 3]
iex> head
1
iex> tail
[2, 3]
```

[head | tail] no solo se usa en la coincidencia de patrones, sino también para anteponer elementos a una lista

```
iex> list = [1, 2, 3]
[1, 2, 3]
iex> [0 | list]
[0, 1, 2, 3]
```

Operador pin

Usa el operador de pin `^` cuando desee comparar patrones con el valor de una variable existente en lugar de vo

```
...
iex> x = 1
1
iex> ^x = 2
** (MatchError) no match of right hand side value: 2
iex> {y, ^x} = {2, 1}
{2, 1}
iex> y
2
...
```

Más sobre pattern matching

Si **una** variable **se** menciona más de **una** vez **en** un patrón, todas las referencias deben unirse **al** mismo patrón

```
...
iex> {x, x} = {1, 1}
{1, 1}
iex> {x, x} = {1, 2}
** (MatchError) no match of right hand side value: {1, 2}
...
```

En algunos casos, uno o varios valores **en** particular **no** son importantes **en** un patrón. Es **una** práctica **común** v

```
...
iex> [head | _] = [1, 2, 3]
[1, 2, 3]
iex> head
1
...
```

Aunque pattern matching nos permite realizar operaciones muy potentes, su uso es limitado. Por ejemplo, **no** pu

```
...
iex> length([1, [2], 3]) = 3
** (CompileError) iex:1: cannot invoke remote function :erlang.length/1 inside match
...
```

