

# Cómo conocí a la metaprogramación (y sobreviví a ello)

Miguel E. Ruiz

29 de febrero de 2024

# Contenidos

- 1 Introducción
- 2 Metaprogramación
- 3 Mi TFM: LogicElixir

# IO.puts("Hello, World!")



- Miguel [Emilio]
- Ingeniero Informático
- Intereses: Programación funcional y concurrente
- Erlang y Elixir

# La metaprogramación y yo

- Al acabar el Grado (2018), interés en Elixir
  - Ejercicios en Codewars
  - Proyecto Pokedex
- Interés en el libro *Metaprogramming Elixir*
- Sin embargo...
  - Falta de tiempo (y de motivación)
- Arranca el máster (2021) y...
  - recupero el interés en la metaprogramación
  - posibilidad de hacer el TFM sobre ello

¿KLK?

# Metaprogramación

Definicion(es)

## Elixir School

*Metaprogramming is the process of using code to write code.*

# Metaprogramación

Definicion(es)

## Elixir School

*Metaprogramming is the process of using code to write code.*

## Rust Web Programming

*Metaprogramming can generally be described as a way in which the program can manipulate itself based on certain instructions.*

# Metaprogramación

Definicion(es)

## Elixir School

*Metaprogramming is the process of using code to write code.*

## Rust Web Programming

*Metaprogramming can generally be described as a way in which the program can manipulate itself based on certain instructions.*

## ChatGPT

*Metaprogramming is a programming technique where a program can manipulate its own structure or behavior at runtime.*



# Metaprogramación

## Técnicas

- Metaclases (Ruby, Python)
- Macros (Scala, Rust)
- Templates (C, C++)

# Metaprogramación en Elixir

## Mi definición

- Metaprogramación: mecanismo para generar programas a partir de otros programas (en tiempo de compilación).
- Elixir tiene la capacidad de tratar los programas como un tipo de dato y modificarlos o transformarlos.
- Ejemplos de uso:
  - Extensión del lenguaje a través de DSL.
  - Creación de nuevas librerías como *Phoenix* y *Ecto*.

# Metaprogramación en Elixir

¿Cómo?

- A través de macros, las cuales permiten **expandir** el lenguaje

¿Cuántas palabras reservadas hay en Elixir?

- 8
- 15
- 37
- ninguna

# Metaprogramación en Elixir

- A través de macros, las cuales permiten **expandir** el lenguaje

¿Cuántas palabras reservadas hay en Elixir?

- 8
- 15
- 37
- ninguna

`true`, `false`, `nil`, `when`, `and`, `or`, `not`, `in`  
`fn`, `do`, `end`, `catch`, `rescue`, `after`, `else`

# Metaprogramación en Elixir

- Con el operador `defmacro` podremos definir nuestras macros :D respetando las reglas
  - Don't Write macros (WTF)
  - Use Macros Gratuitously

# AST (de Elixir)

El AST es un tipo de dato de Elixir que se representa mediante una tupla de 3 elementos:

- Un átomo que representa la función invocada u otra tupla que representa un nodo interno del árbol.
- Una lista de términos de metadatos de la función invocada.
- Una lista de argumentos de la función del primer argumento.

# Programación funcional

## AST de Elixir

Para obtener el AST de una expresión de Elixir, se utiliza la función `quote/2`:

```
iex> quote do: sum(1, 2, 3)
{:sum, [], [1, 2, 3]}
```

Si se requiere introducir valores dentro de un AST, se dispone de la función `unquote/1`:

```
iex> a = 3
iex> quote do: sum(1, 2, a)
{:sum, [], [1, 2, {:a, [], Elixir}]}
iex> quote do: sum(1, 2, unquote(a))
{:sum, [], [1, 2, 3]}
```

# DEMO `macro unless`, `macro defpet` (por definir)



- cosas a traer de la presentación del TFM

# Bibliografía