



DIGITAL
NATIVES

INTRODUÇÃO A ELM: PROGRAMAÇÃO FUNCIONAL NO FRONT-END

RODRIGO NONOSE

MARINGÁ, 11. MARÇO. 2017.

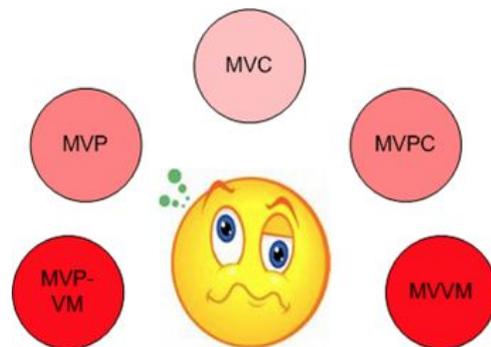
Sumário

- Introdução
 - Contexto
 - Problema
 - Solução (?)
- Elm
 - Programação Funcional
 - Syntaxe Básica
 - Arquitetura
 - Tooling

Introdução

Contexto

- HTML, Javascript e CSS
- Frameworks:
 - Gerenciar estado
 - Promessas
 - MVC, MVVM, MVP, MVVMCMPSCPMSC
- Arquitetura em camadas para arquitetura modular
 - React, Redux
- Two-way binding
- One-directional binding
- Virtual DOM



Introdução

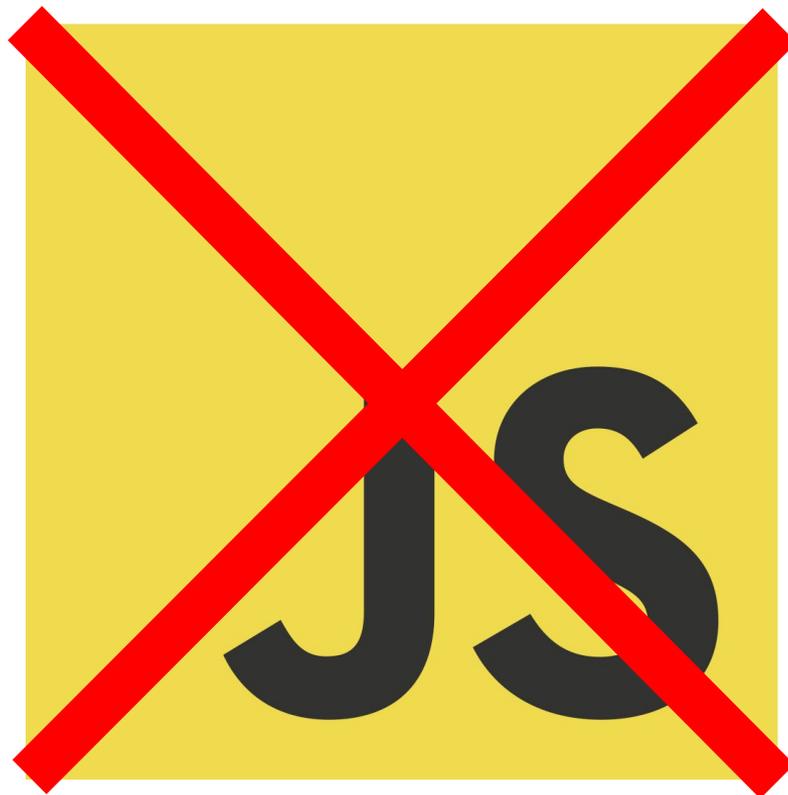
Problema



JS

Introdução

Solução(?)





Elm

Programação funcional

- Linguagem funcional compilada para javascript
 - Compilador escrito em Haskell
- Estaticamente e fortemente tipada
 - Systema de tipo Hindley-Milner
- Expressividade, conciso e autodocumentado
 - Corpo da função consiste em um único *statement*
 - Uma falha de compilação sem uma mensagem clara é considerado bug
- Imutabilidade e Transparência Referencial
 - Funções não têm efeito colateral
 - Tipos imutáveis



Elm

Programação funcional



Gregory Schier

@GregorySchier



Thanks [@elmlang](#) for the most useful error message I've ever seen

```

Detected errors in 1 module.

-- ALIAS PROBLEM ----- ../Maze.elm

This type alias is recursive, forming an infinite type!

21 |> type alias Node =
22 |>   { x : Int
23 |>   , y : Int
24 |>   , children : List Node
25 |>   }

When I expand a recursive type alias, it just keeps getting bigger and bigger.
So dealiasing results in an infinitely large type! Try this instead:

    type Node
      = Node { x : Int, y : Int, children : List Node }

This is kind of a subtle distinction. I suggested the naive fix, but you can
often do something a bit nicer. So I would recommend reading more at:
<https://github.com/elm-lang/elm-compiler/blob/0.17.0/hints/recursive-alias.md>
  
```



John Carmack ✓

@ID_AA_Carmack



That should be an inspiration for every error message.



Gregory Schier @GregorySchier

Thanks [@elmlang](#) for the most useful error message I've ever seen

Elm



I call it my billion-dollar mistake. It was the invention of the null reference in 1965.

— *Tony Hoare* —

Elm

Programação funcional

- Não existe null
 - Sempre lidar com as duas possibilidades utilizando um wrapper: Maybe
- Não existe undefined function
 - Detectado em tempo de compilação

Elm

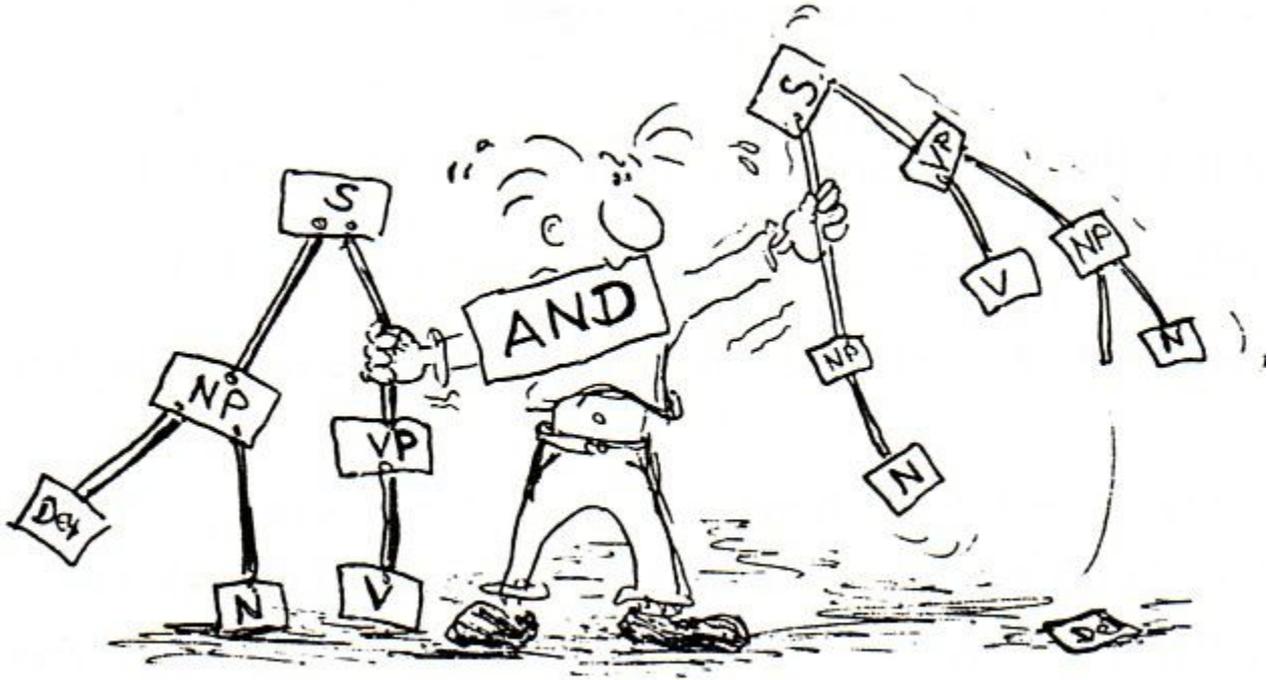
Programação funcional

- Não existe null



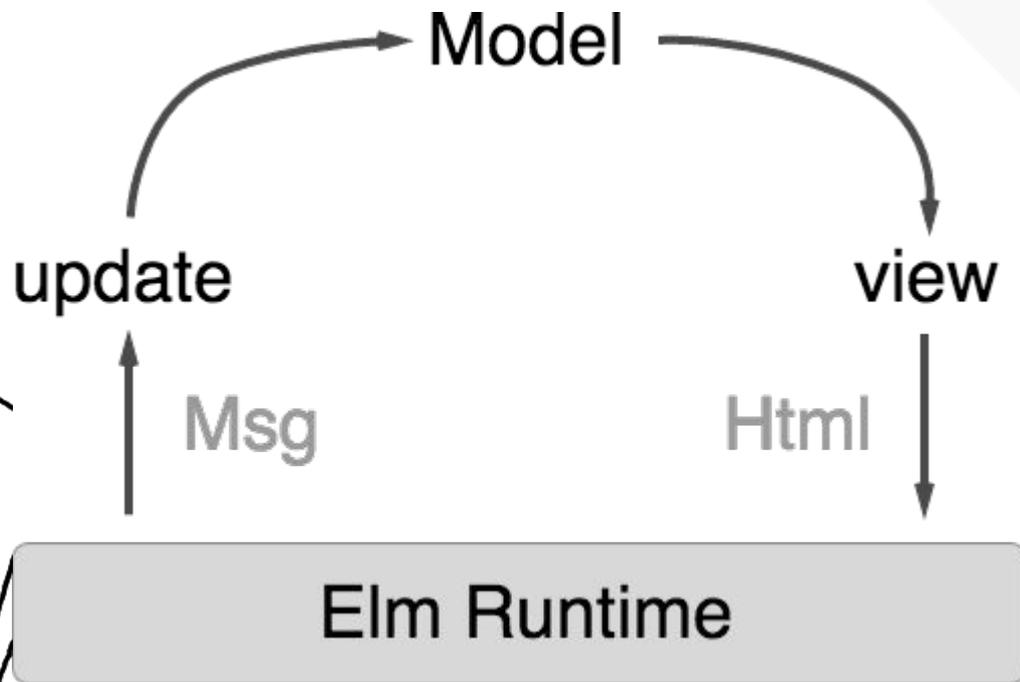
Elm

Syntaxe Básica



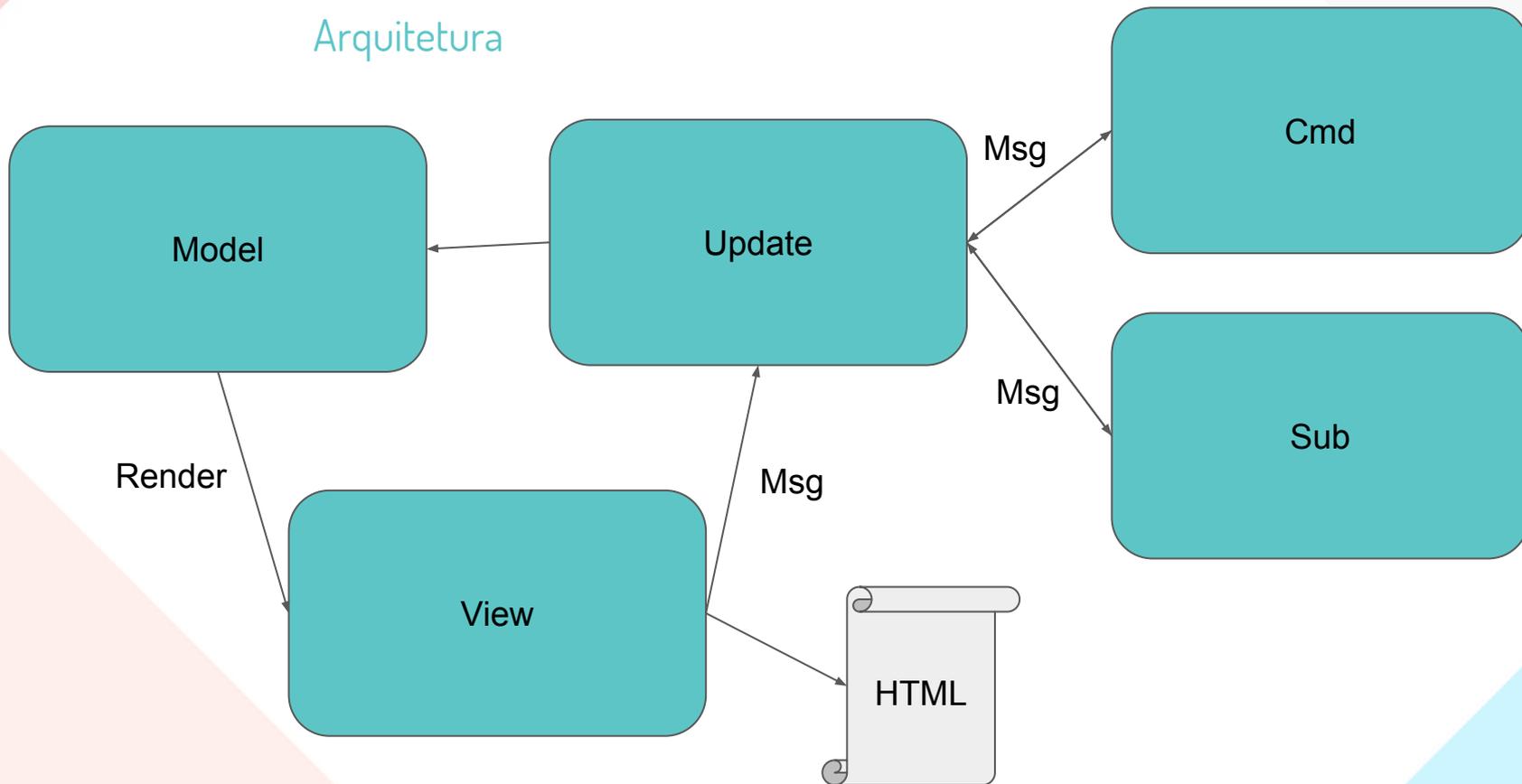
Elm

Arquitectura



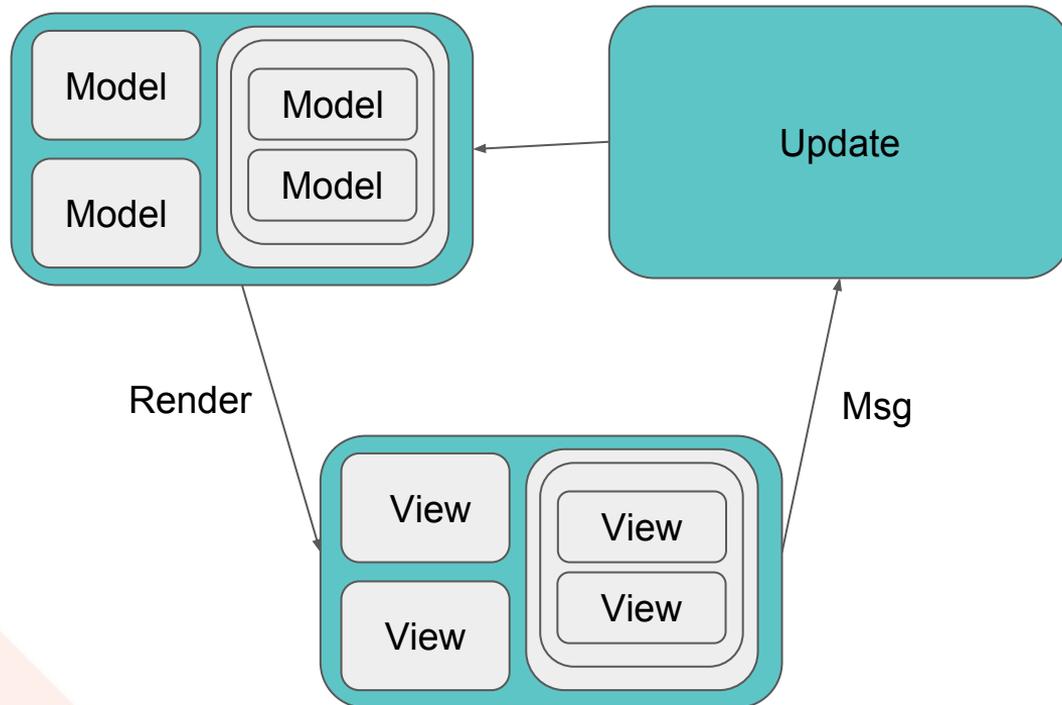
Elm

Arquitectura



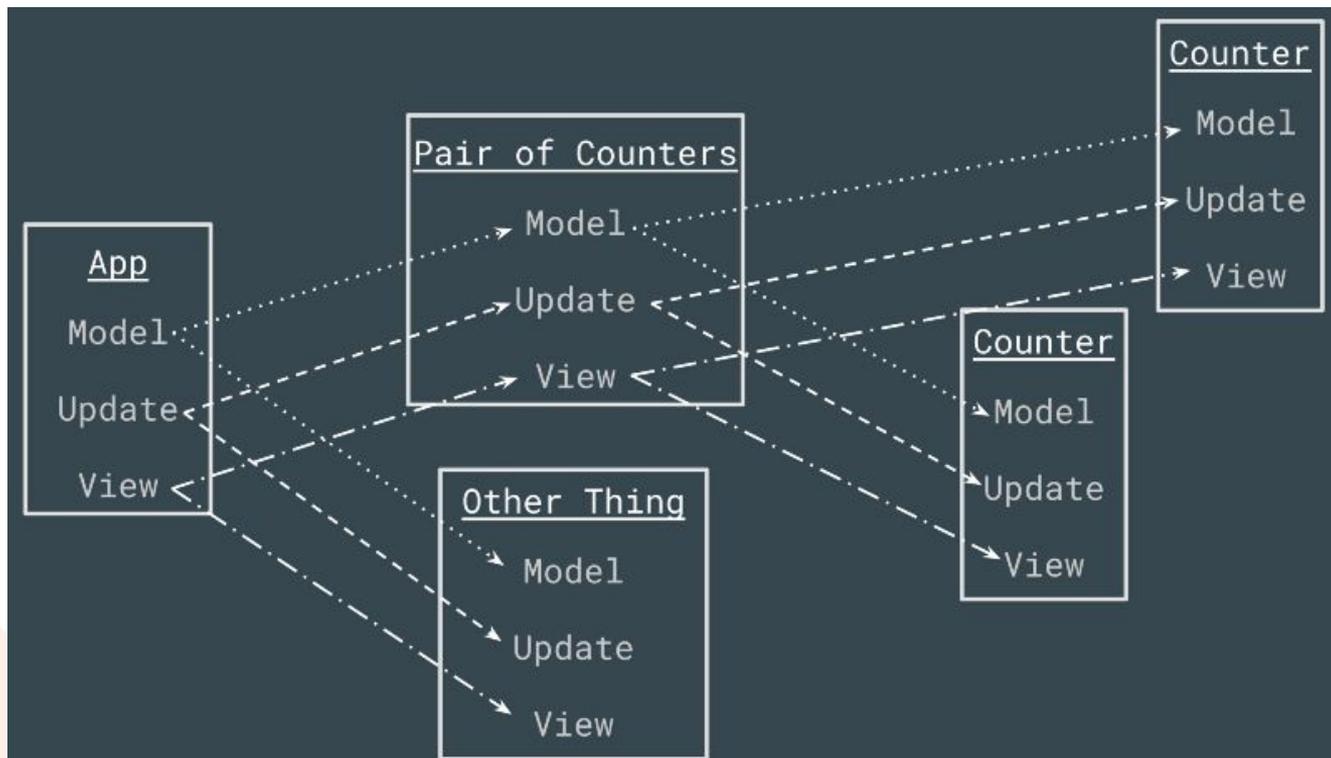
Elm

Arquitectura



Elm

Arquitectura



Elm

Tooling

- Compilador com mensagens de erro amigáveis
- Elm Reactor: serve os arquivos e code-reload
- Elm Formatter
- Elm Package: previne quebra de versionamento semântico
- Debugger: Time Travel

Elm

Tooling

- Compilador com mensagens de erro amigáveis
- Elm Reactor: serve os arquivos e code-reload
- Elm Formatter
- Elm Package: previne quebra de versionamento semântico
- Debugger: Time Travel



S.B



FLW VLW