

UM () AOS CALLBACKS

CSP, implementado em Clojure

Programação Concorrente

Computações que podem ser executadas
em paralelo e comunicar entre si

Sistemas Distribuídos

Interfaces Gráficas

ou seja,

QUASE TUDO

Computação Paralela

I/O



2 caminhos*

**A fork in the road*

A dramatic, apocalyptic scene of a city engulfed in flames and smoke, with lightning striking the sky. The image is a high-contrast, fiery landscape with a dark, stormy sky. The city below is a dense collection of buildings, many of which are on fire, with bright orange and yellow flames rising from the streets and rooftops. Thick, dark smoke billows from the burning structures, filling the air. In the background, a prominent, tall, spire-like building stands out against the dark sky, with a bright light emanating from its peak. Several bright, jagged lightning bolts strike down from the dark, cloudy sky, illuminating the scene with a harsh, orange glow. The overall atmosphere is one of intense destruction and chaos.

Partilha de memória

Troca de mensagens

A man with dark hair, wearing a red and blue plaid shirt, stands in a field of tall, golden-brown grass. He is smiling and has his arms around a young girl on his left and a young boy on his right. The girl is wearing a bright pink shirt and is also smiling. The boy is wearing a grey button-down shirt and is smiling. The background shows a clear blue sky and some distant trees on the horizon.

2 modelos

+

1 coisa a evitar

#1

(ver Scala, Erlang ou o nosso Galeno!)

Atores

#2

(Aquilo que vamos ver)

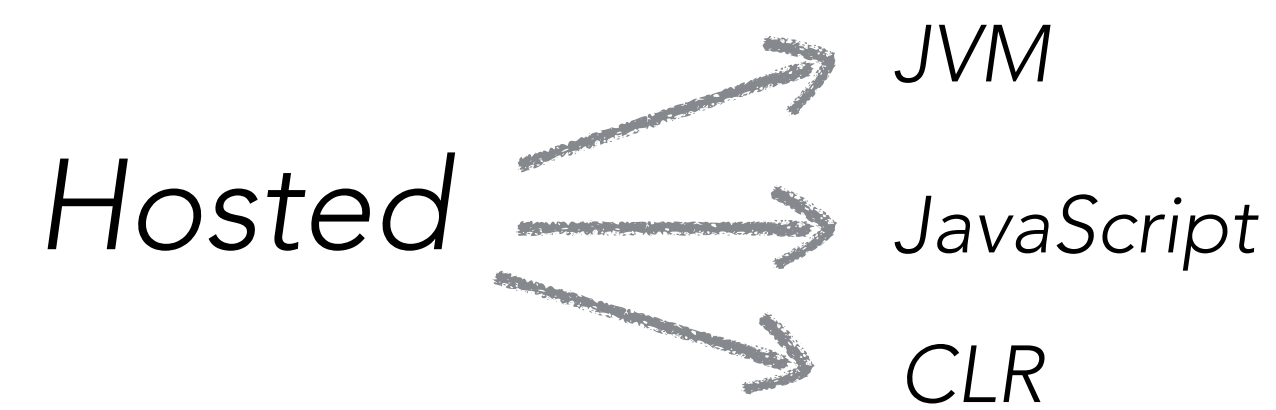
CSP

A coisa a evitar

Inversão de Controlo



Clojure



LISP



Dinâmico

*Polimorfismo
em Runtime*

Concorrência

Clojure

Funcional

Identidade != Estado

Tema para outra talk...



Alguns tipos básicos

Lista

`(1, 2, 3)`

Vetor

`[1, 2, 3]`

Mapa

`{:a 1, :b 2, :c 3}`

A vírgula é whitespace.



Ligar um valor a um símbolo

```
(def a [1 2 3])
```




Definir uma função

Função anónima `(fn [x] ...)`

Função com nome `(defn a [x] ...)`



Chamar uma função

Notação pré-fixa

ZOMG!!1!

Passa de (JS): `f (1, 2, 3) ;`

Para: `(f 1 2 3)`



Outras coisas importantes

Binding Léxico `(let [x 1] ...)`

Recursão `(loop [x 1] ...
 (recur (inc x)))`

core.async

Let's look at the code...



/dzacarias/pm-async