***Aula 8-8***

Hue br

https://github.com/MostlyAdequate/mostly-adequate-guide-pt-BR

Livros –-> Function Javascript

***Criar uma pasta mkdir pasta***

***Instalar o node :***

***sudo apt-get install -y nodejs***

***Instalar :***

***Ubuntu***

***npm install ramda***

var max = function (a,b) {

if (a > b )

return a;

else

return b

}

max (12,15)

max (-1,-2)

//Criando objetos

var medal = { type : “gold”, winner: “rafaela”, sport: “judo”}

medal.type

medal.winner

medal.sport

medal.date=”8AGO”

require(“ramda”)

//ele acessou o modo que esta ai da pasta de vocês, ele está acessível.

Var \_ = require(“ramda”) var curry = require(“ramda”).curry

var max2 = \_.curry(max) var max2 = curry(max)

var dez\_ou = max2(10)

dez\_ou(5)

dez\_ou(15)

\_ = require(“ramda”)

dez\_ou(5)

dez\_ou(15)

Var \_\_ = require(“ramda”)

\_\_.match(/ \ s + /g, “hello World”) // retorna um espaço[ ' ' ]

\_\_.match((/ \ s + /g,”olympics”) // retorna nada []

var hasSpaces = \_\_.match(/\s+/g) // Aplicação parcial

hasSpaces(“Hello Word”) // retorna também um espaço

\_\_.map( function(x) { return x+1}, [2,3,4,5]) // aplica a função em todos os elementos

\_\_.map( function(x){ return 2\*x}, [2,3,4,5])

\_\_.map( function(x){return x%2}, [2,3,4,5])

\_\_.map( funciton(x){return x%2 == 0},[2,3,4,5])

\_\_.filter(function(x){return x%2 == 0},[2,3,4,5])

// mostra a onde esta aplicando onde , essa função retorna = 0

var pares = \_\_.filter(function(x) {return x%2 == 0})

pares ( [0,1,10])

BAIXAR O ZIP

https://github.com/MostlyAdequate/mostly-adequate-guide

MV ~/Downloads/mostrl.... ~/ncn

cd ~/ncn

unzip mosty...

cd mostly

cd code

cd part1\_exercises

npm install

npm install -g

npm test

node\_modules/mocha/bin/mocha exercises/curry/curry\_exercises\_spec.js -w

node\_modules/.bin/mocha

abrir atom exercises/curry/curry\_exercuses.js

CAPITULO 3

SLICE É PURO POIS É O MESMO RESULTADO

splice cada hora vem uma coisa

Então essa é a essência da programação funcional : ser previsível

Efeito colaterais – > modificar um arquivo de sistema -> o que importa da sua função é seu retorno,

alterar estado em uma variável

Isso não é proibido, isto vai está separado!!!Isso não vai ser uma coisa coxa que fazemos a qualquer hora

var memoize = function(f) {

var cache = {};

return function() {

var arg\_str = JSON.stringify(arguments);

cache[arg\_str] = cache[arg\_str] || f.apply(f, arguments);

return cache[arg\_str];

};

};

IR NO LIVRO

COMPOSIÇÂO DE FUNÇÔES DE ORDEM SUPERIOR

var compose = function (f,g) {

return function (x){

return f(g(x));

}

}