

# Programação funcional

---

Marco A L Barbosa  
malbarbo.pro.br

Departamento de Informática  
Universidade Estadual de Maringá



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-CompartilhaIgual 4.0 Internacional.  
<http://github.com/malbarbo/na-progfun>

**O quê?**

# O quê?

- É um paradigma de programação inspirado no cálculo lambda
- A computação é especificada pela definição e aplicação de funções (programação imperativa é baseado em mudança de estado)
  - Evita dados mutáveis

**Por quê?**

## Por quê?

- Um paradigma (linguagem) de programação é uma ferramenta
- Conhecer várias ferramentas permite utilizar a mais adequada para cada problema
- Programas mais simples, fáceis de entender e modificar
- Programas mais fáceis de paralelizar (melhorar a utilização do hardware atual)
- Diversão! Programar (no paradigma funcional) é muito divertido

**Como?**

# Como?

- Escolher uma linguagem
- Ler a fundamentação teórica e as referências da linguagem
- Praticar lendo e escrevendo código

# Como?

- Escolher uma linguagem
  - Racket
  - Bom suporte ao paradigma funcional
  - Variante moderna do Lisp
  - Ambiente integrado DrRacket
  - Documentação extensa
  - Fácil instalação



# Como?

- Ler a fundamentação teórica e as referências da linguagem
  - A Tutorial Introduction to the Lambda Calculus
  - Livro How to Design Programs
  - Guia e Referência do Racket
  - Livro Structure and Interpretation of Computer Programs
  - Livro The Scheme Programming Language

# Como?

- Praticar lendo e escrevendo código
  - Muitos exemplos
  - Muitos exercícios

## **Primeiros passos**

```
$ apt-get install racket
```

```
$ drracket
```

Ficheiro Editar Ver Linguagem Racket Insert Scripts Tabs Ajuda

SemNome 2 (define ...) Verificar Sintaxe Debug Macro Stepper Executar Parar

```
#lang racket
```

Bem-vindo a [DrRacket](#), versão 7.2 [3m].

Linguagem: racket, with debugging; memory limit: 128 MB.

```
> |
```

**Selecione esta opção**

Determine language from source

3:2

559.93 MB



```
#lang racket
```

```
(define (quadrado x)  
  (* x x))
```

## Definições

Bem-vindo a [DrRacket](#), versão 7.2 [3m].

Linguagem: racket, with debugging; memory limit: 128 MB.

```
> (quadrado 5)
```

```
25
```

```
>
```

## Interações

- Na área de definições
  - Digite o código do programa
  - Pressione o botão executar/correr (`ctrl` + `r`)



- Na área de interações
  - Teste as definições usando REPL (Read Eval Print Loop)
    - Entre com uma expressão (Read)
    - A expressão é avaliada (Eval)
    - O resultado da expressão é impressa (Print)
    - Repetida o processo (Loop)

- Exemplos de interações

```
> 34
```

```
34
```

```
> (+ 3 4)
```

```
7
```

```
> (* 2 6)
```

```
12
```

```
> +
```

```
#<procedure:+>
```

**Leitura recomendada**

- Introdução rápida ao Racket
- The Python paradox
- Programação funcional
- Revenge of the Nerds
- Beating the averages