



Disciplina: **AARE Paradigmas de Linguagens de Programação**

Professor: Prof. Ausberto S. Castro V.

E-mail: ascv@uenf.br

Data: 24 de setembro de 2022

## Prática – Racket

Nome Completo: Ricardo Willian Pontes da Silva

Data: 24 de setembro de 2022

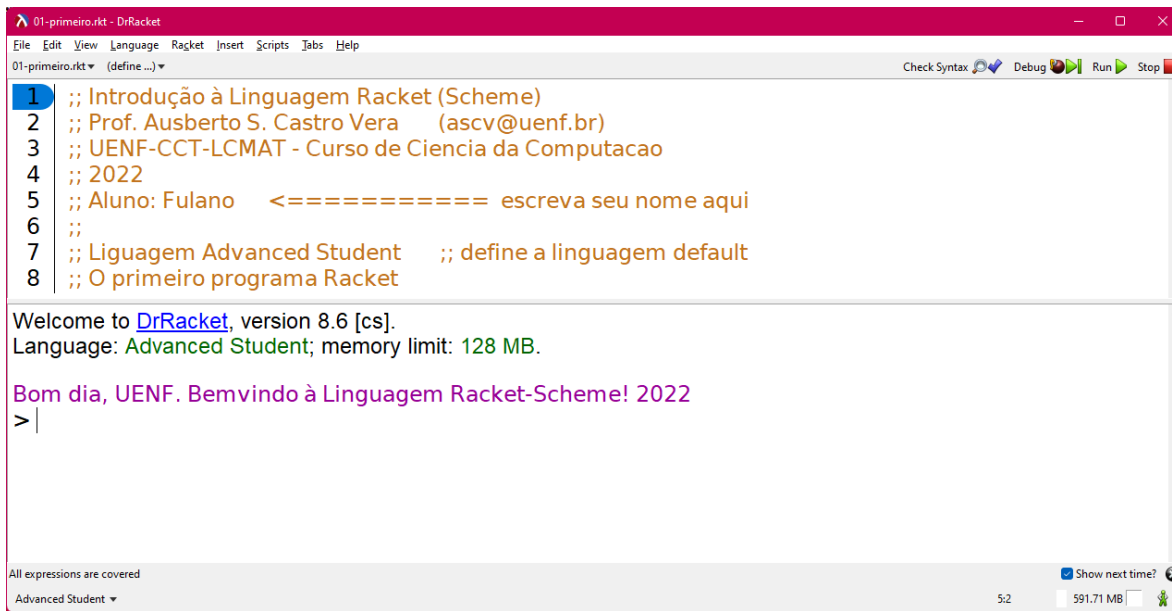
Total Exercícios Resolvidos: .....

Arquivo 01-primeiro.rkt **Primeiro programa em Racket**

1. Execute o programa e indicar o que faz cada linha do código fonte do programa. Quais funções estão definidas nas 4 linhas. Explique cada uma delas?

```
1 ;; Introdução à Linguagem Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciência da Computação
4 ;; 2022
5 ;; Aluno: Fulano <===== escreva seu nome aqui
6 ;;
7 ;; Linguagem Advanced Student ;; define a linguagem default
8 ;; O primeiro programa Racket
9 ;; -----
10 (begin
11   (newline)
12   (display "Bom dia, UENF. Bemvindo à Linguagem Racket-Scheme! 2022") ;<----- mostrar na
12  tela
13   (newline))
14
15
```

Código fonte



Interações (shell)

Explicar aqui:

```

;; Introdução à Linguagem Racket (Scheme)
;; Prof. Ausberto S. Castro Vera (ascv@uenf.br)
;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
;; 2022
;; Aluno: Ricardo Willian Pontes da Silva
;;
;; Liguagem Advanced Student ;; define a linguagem default
;; O primeiro programa Racket
;; -----
(begin
  (newline)
  (display "Bom dia, UENF. Bemvindo à Linguagem Racket-Scheme! 2022") ;<----- mostrar na tela
  (newline)
)

```

**begin:** A função begin será utilizada para indicar o início do programa.

**newline:** Será utilizada toda vez que se desejar acrescentar uma linha em branco na janela de execução.

**display:** A função display será utilizada para expressar no dispositivo de saída alguma mensagem.

O programa tem como objetivo imprimir no dispositivo de saída (monitor) a mensagem “bom dia, UENF, bem vindo à linguagem Racket-Scheme! 2022”

2. Agregar linhas de código para mostrar na parte executável, a mensagem “Pratica 01 – Linguagem Racket”, o nome completo do aluno e a data atual

```

1 ;; Introdução à Linguagem Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; 2022
5 ;; Aluno: Fulano Ricardo Willian Pontes da Silva
6 ;;
7 ;; Liguagem Advanced Student          ;; define a linguagem default
8 ;; O primeiro programa Racket
9 ;; -----
10 (begin
11   (newline)
12   (display "Bom dia, UENF. Bemvindo à Linguagem Racket-Scheme! 2022")
13   (display "Pratica 01 - Linguagem Racket")
14   (display "Ricardo Willian Pontes da Silva")
15   (display "21/09/2022")
16   (newline))
17
18

```

Arquivo 02-numeros.rtk [Números e Aritmética](#)

3. Execute o programa e mostre os resultados

```

02-numeros.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
02-numeros.rkt (define ...)
1 ;; Introdução à Linguagem Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva

Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
    UENF-CCT-LCMAT-CC, 2022
    Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
    Aluno: Fulano
Soma 23 + 28 = 51
Produto 14*17 = 238
Combinando 5 + (3*7) = 26
Combinando (2 + (3*4))/2 - 4 = 3
Raiz quadrada de 4= 2
Raiz quadrada de 2= #i1.4142135623730951
Complexos - raiz quadrada de -1= 0+1i


Valor de Pi+1 #i4.141592653589793
Seno 90 graus: #i1.0
Coseno 60 graus: #i0.5000000000000001
Coseno 45 graus: #i0.7071067811865476
Logaritmo Natural de 15: #i2.70805020110221
exponente 2^3 = 8
exponente 4^(1/2) = 2
Maximo de 1 3 4 2 3 = 4
minimo de 1 3 4 2 3 = 1
valor absoluto de 3 = 3
valor absoluto de -4 = 4

"quotes obriga as listas serem tratadas como DADOS"
(list 2 4 6)
(list '/ 4 (list '* 3 7))
(list 1 2 3 4)
(list (list 'a 'b) (list 3 5))
(list '+ 2 (list '* 5 7))
>

```

4. Escreva programas Racket para as seguintes expressões:
  - 4.1.  $H = (4 - (7^2 + 6^3) / 3) - (6 + (5 - (2^4 - 8)))$
  - 4.2. Escreva um NOVO programa Racket que calcule o valor da expressão:

$$\frac{\sqrt{7^2 + 5 + \sin(18 - 7) + \cos(20 + 2)}}{(5 + 3) * (4 - 8)^2}$$

 **exercício 4.1** - DrRacket


File Edit View Language Racket Insert Scripts Tabs Help

exercício 4.1 ▾ (define ...) ▾


1: 03-variaveis.rkt × 2: exercício 4.1 × +

```
1 ;; Introdução à Linguagem Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 ;; Liguagem Advanced Student          ;; define a linguagem default
8 (begin
9
10     (display (- (- 4 (/ (+ (expt 7 2) (expt 6 3)) 3)) (+ 6 (- 5 (- (expt 2 4) 8)))))
11
12
13 )
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: **Advanced Student**; memory limit: 128 MB.  
**-262/3**  
>

 **exercício 4.2.rkt** - DrRacket\*

File Edit View Language Racket Insert Scripts Tabs Help

exercício 4.2.rkt ▾ (define ...) ▾ 

1: 03-variaveis.rkt × ★ 2: exercício 4.2.rkt × +

```
1 ;; Introdução à Linguagem Racket (Scheme)
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 ;; Liguagem Advanced Student          ;; define a linguagem default
8 (begin
9
10     (display (/ (sqrt ((+ ((+ 49 5) (sin (- 18 7))) (cos (+ 20 2)))) (* (+ 5 3) 16))
11
12 )
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: **Advanced Student**; memory limit: 128 MB.  
**0.05633676520454323**  
>

Arquivo [03-variaveis.rkt](#)

5. Execute o programa e indique o valor das variáveis m, z, k
  - 5.1. Explicar o significado de cada uma das 3 expressões de iteração `let`
  - 5.2. Escreva 2 expressões do tipo `let` e explique o seu significado

```

9 | #lang racket           ;; define a linguagem default
10 | ; -----
11 | (display "  UENF-CCT-LCMAT-CC, 2022")
12 | (newline)
13 | (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
14 | (newline)
15 | (display "  Aluno: Ricardo Willian Pontes da Silva ")
16 | (newline)
17 | ;;
18 | ;;
19 | ;; Variáveis e Expressões LET
20 | ;; -----
21 | (define x 3)                ; para x=3
22 | (define m 7)
23 | (define z (+ m 6))          ; para z = m+ 6
24 | (define Pi 3.141516)
25 | (define k (+ z (- m x)))
26 |
27 | (define (quadrado x)         ; (define (nomeFuncao parametro) definicao )
28 |   (* x x) )
29 |
30 |
31 | (define (entre5e20? n)
-- | -----

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

O numero 8 esta entre 5 e 20? #t

O numero 37 esta entre 5 e 20? #f

Escreva qualquer variavel definida acima ... <var> <ENTER> x m z Pi

30

43

23

O quadrado de 12 : 144

Valor de m: 7

Valor de z: 13

Valor de k: 17

### 5.1

A expressão `let` é utilizada para restringir variáveis locais em uma subexpressão, esse conceito é bastante utilizado em linguagens funcionais.

## 5.2

Untitled 2 - DrRacket\*

File Edit View Language Racket Insert Scripts Tabs Help

Untitled 2 ▾ (define ...) ➡

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da
6
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ; -----
9 (define x 0)
10 (begin
11
12   (let ((x 5))
13     (* x x))
14
15 )
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: [Advanced Student](#); memory limit: 128 MB.

25  
>

Arquivo [04-areas.rtk](#)

6. Execute o programa e indique o que faz o programa

```
04-areas.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
04-areas.rkt (define ...)

15 (newline)
16 (display " Aluno: Ricardo Willian Pontes da Silva")
17 (newline)
18 (newline)
19 ;;
20 ;;
21 ;; (define (nomeFuncao parametros) definicao )
22 ;;
23 ;; AREAS
24 ;; -----
25 (define (area-circulo r)
26   (* 3.14 (* r r))
27 )
28 (define (areatriangulo b h)
29   (/ (* b h) 2))
30
31 (define (areadisco interno externo)
32   (- (area-circulo externo)
33      (area-circulo interno))
34 )

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Ricardo Willian Pontes da Silva

Circulo de raio 18 AREA = 1017.36
Triangulo base=10, altura=15 AREA = 75
Disco raio menor=12, raio maior=20 AREA = 803.8399999999999
>
```

O programa **04-areas.rtk** tem como objetivo calcular áreas internas de figuras geométricas como círculo e triângulo, mas também calcular a área interna entre dois círculos.

- 6.1. Escreva um programa Racket para calcular a área de um quadrado qualquer, a área de um trapézio e a área de um polígono.




```
exercício 6.1.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
exercício 6.1.rkt (define ...)
1: 04-areas.rkt x 2: exercício 6.1.rkt x +

5 ;; Aluno: Ricardo Willian Pontes da Silva
6
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ;;
9 ; -----
10
11 (define (area-quadrado r )
12   (* r r))
13 (display "area de um quadrado de lado 10 :")
14   (area-quadrado 10)
15
16
17 (define (area-trapezio base_maior base_menor h )
18   (/ (* (+ base_maior base_menor) h ) 2))
19
20 (display "area de um trapezio de base maior = 10, base menor = 2 e altura = 3 :")
21   (area-trapezio 10 2 3)
22
23 (define (area-poligono p a )
24   (* (* 0.5 p) a))
25
26 (display "area de um poligono de perimetro = 120 e apótema = 3:")
27   (area-poligono 120 3)

Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
area de um quadrado de lado 10 :100
area de um trapezio de base maior = 10, base menor = 2 e altura = 3 :18
area de um poligono de perimetro = 120 e apótema = 3:180
> |
```

- 6.2. Escreva um programa NOVO completo para calcular o volume de um galão de óleo utilizando a fórmula  $V = \pi R^2 A$ , onde as variáveis V, R e A representam, respectivamente, o volume, o raio e a altura

 exercicio 6.2.rkt - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

exercicio 6.2.rkt ▾ (define ...) ▾

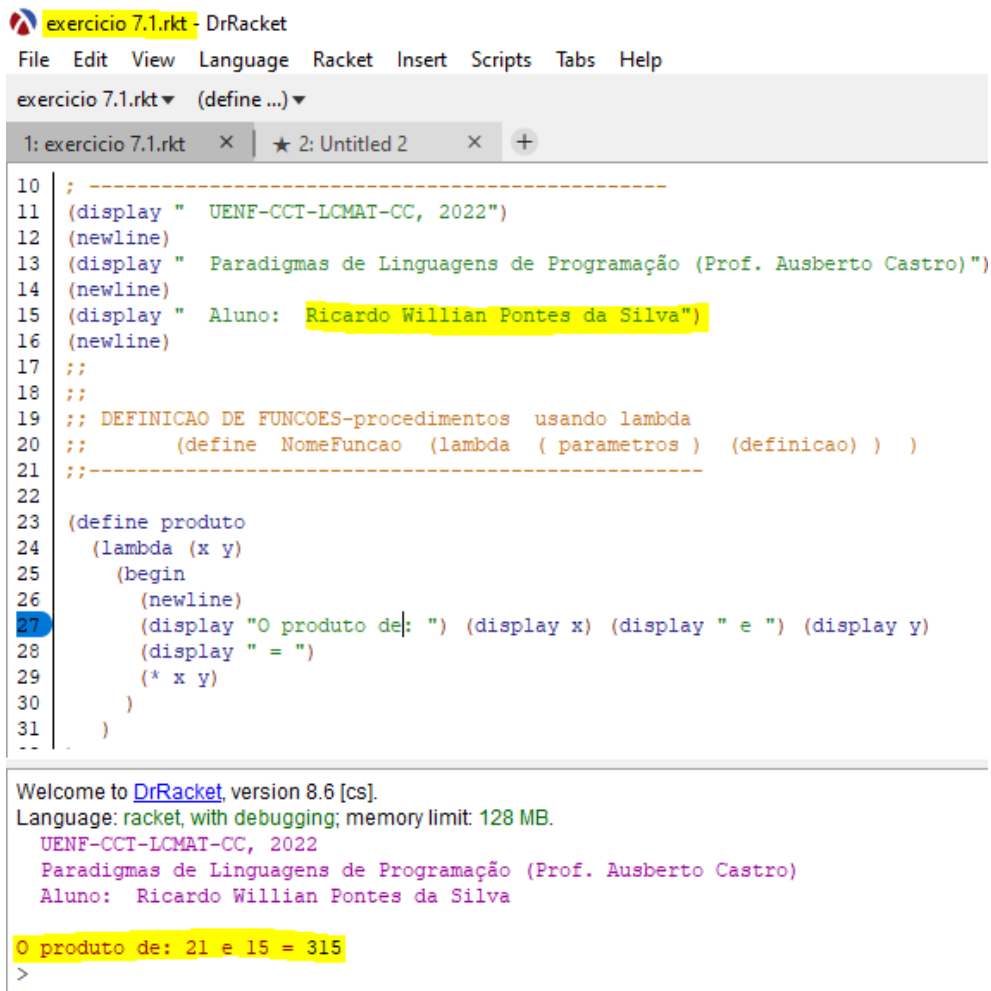
```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6
7 ;;;;;;;;;;;;;; Escolha a linguagem "Determine language from source"
8 ;;
9 ; -----
10
11 (define (volume_galao r a )
12   (*(* r r)a)3.14))
13 (display "O volume de um galão com raio = 5, altura = 10 é igual :")
14   (volume_galao 5 10)
15
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: [Advanced Student](#); memory limit: 128 MB.  
O volume de um galão com raio = 5, altura = 10 é igual :785  
>

Arquivo [05-funcoes.rkt](#)

7. Execute o programa e explique o que faz





The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The title bar indicates the file is 'exercicio 7.1.rkt'. The editor window displays a Racket script with the following code:

```
10 ; -----
11 (display " UENF-CCT-LCMAT-CC, 2022")
12 (newline)
13 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
14 (newline)
15 (display " Aluno: Ricardo Willian Pontes da Silva")
16 (newline)
17 ;;
18 ;;
19 ;; DEFINICAO DE FUNCOES-procedimentos usando lambda
20 ;; (define NomeFuncao (lambda ( parametros ) (definicao) ) )
21 ;;-----
22
23 (define produto
24   (lambda (x y)
25     (begin
26       (newline)
27       (display "O produto de: ") (display x) (display " e ") (display y)
28       (display " = ")
29       (* x y)
30     )
31   )
32 )
```

The bottom output window shows the following text:

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Ricardo Willian Pontes da Silva

O produto de: 21 e 15 = 315
>
```

#### Arquivo [06-condicionalIF.rkt](#)

8. Executar e explicar o programa
  - 8.1. Escreva um programa com dois condicionais
  - 8.2. Escreva um programa para calcular a média de três notas e indique “Aprovado” se for maior ou igual a 6,0, e “Reprovado”, caso contrário

#### Arquivo [07-formulas.rkt](#)

9. Executar e explicar o programa

```
07-formulas.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
07-formulas.rkt (define ...) ➡

4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket ;; define a linguagem default
8 ; -----
9 (display " UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display " Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Ricardo Willian Pontes da Silva

Pitagoras de 3 e 4 = 25

Aproximando Pi com 4 =  $3\frac{107}{315}$ 


Aproximando Pi com 10 =  $3\frac{3380087}{14549535}$ 

Factorial de 5 = 120

Fazer outro teste sobre Pitagoras e outro sobre Fatorial
~
```

Assim como todas as linguagens de programação, o scheme/racket possui funções pré definidas que auxiliam no decorrer do código, como a função Pitágoras, Factorial e dentre muitas outra como citado no código acima.

- 9.1. Escreva um programa que calcule o fatorial de um número de uma forma diferente da apresentada.

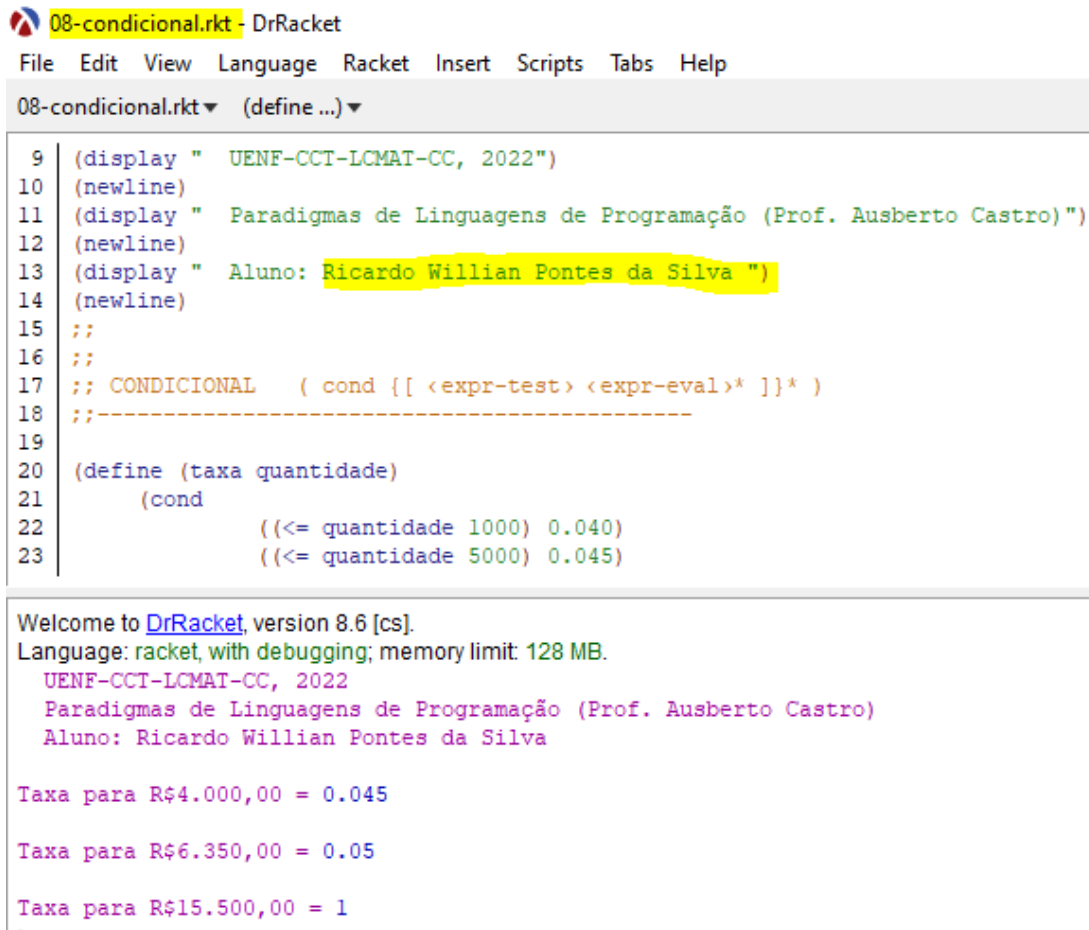
 **exercicio 9.1.rkt** - DrRacket  
File Edit View Language Racket Insert Scripts Tabs Help  
exercicio 9.1.rkt ▾ (define ...) ▾  
1: 07-formulas.rkt × | 2: exercicio 9.1.rkt × +  

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (define (fatorial n)
10   (if (= n 0)
11       1
12       (* n (fatorial (- n 1)))))
13 (fatorial 5)
```

  
Welcome to [DrRacket](#), version 8.6 [cs].  
Language: [racket](#), with [debugging](#); memory limit: 128 MB.  
120

Arquivo [08-condicional.rtk](#)

10. Execute o programa e indique o que faz



```
9 (display " UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display " Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 ;;
16 ;;
17 ;; CONDICIONAL (cond [[ <expr-test> <expr-eval>* ]]*)
18 ;;-----
19
20 (define (taxa quantidade)
21   (cond
22     ((<= quantidade 1000) 0.040)
23     ((<= quantidade 5000) 0.045)
24     (else 1)))
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: Ricardo Willian Pontes da Silva

Taxa para R\$4.000,00 = 0.045

Taxa para R\$6.350,00 = 0.05

Taxa para R\$15.500,00 = 1

Os condicionais permitem que um programa execute diferentes comandos de acordo com as condições estabelecidas. No trecho de código acima podemos ver uma situação em prática, que será comparado três valores e assim será impresso uma variável de acordo com essa condição.

10.1. Escreva um programa condicional com pelo menos 5 opções

```

1  ;; Introdução à Linguagem Scheme-Racket
2  ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; Abril 2022
5  ;; Aluno: Ricardo Willian Pontes da Silva
6  ;;
7
8  ; -----
9  (define (calendario dia)
10     (cond
11         ((equal? dia "Segunda") "Dia útil")
12         ((equal? dia "Terça") "Dia útil")
13         ((equal? dia "Quarta") "Dia útil")
14         ((equal? dia "Quinta") "Dia útil")
15         ((equal? dia "Sexta") "Dia útil")
16         ((equal? dia "Sábado") "Final de semana")
17         ((equal? dia "Domingo") "Final de semana")
18         (else "Opção inválida")
19     )
20 )
21
22 (calendario "Sábado")

```

Welcome to [DrRacket](#), version 8.6 [cs].


Language: [Advanced Student](#); memory limit: 128 MB.

"Final de semana"

>

- 10.2. Escreva um programa `bhaskara.rkt` que calcule as raízes de uma equação  $25x^2 - 55x + 10 = 0$ , utilizando a fórmula de Bhaskara. Sugestão: Primeiro faça o algoritmo completo



 **exercicio 10.2.rkt** - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

exercicio 10.2.rkt ▾ (define ...) ▾

1: exercicio 10.1.rkt × | 2: exercicio 10.2.rkt × +

```
1 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
2 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
3 ;; Abril 2022
4 ;; Aluno: Ricardo Willian Pontes da Silva
5 ;;
6
7 ; -----
8 (define (bhaskara a b c)
9   (local ((define delta (- (* b b) (* 4 a c))))
10    (cond
11      [(< delta 0) empty]
12      [else
13       (local ((define x1 (/ (- (- b) (sqrt delta)) (* 2 a)))
14              (define x2 (/ (+ (- b) (sqrt delta)) (* 2 a))))
15        (cond
16          [(= delta 0) (list x1)]
17          [else (list x1 x2)]))]))))
18 (bhaskara 25 -55 10)
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: **Advanced Student**; memory limit: 128 MB.  
**(list 0.2 2)**  
> |

Arquivo [09-pares.rtk](#)

11. Execute o programa

```
09-pares.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
09-pares.rkt (define ...) ➡

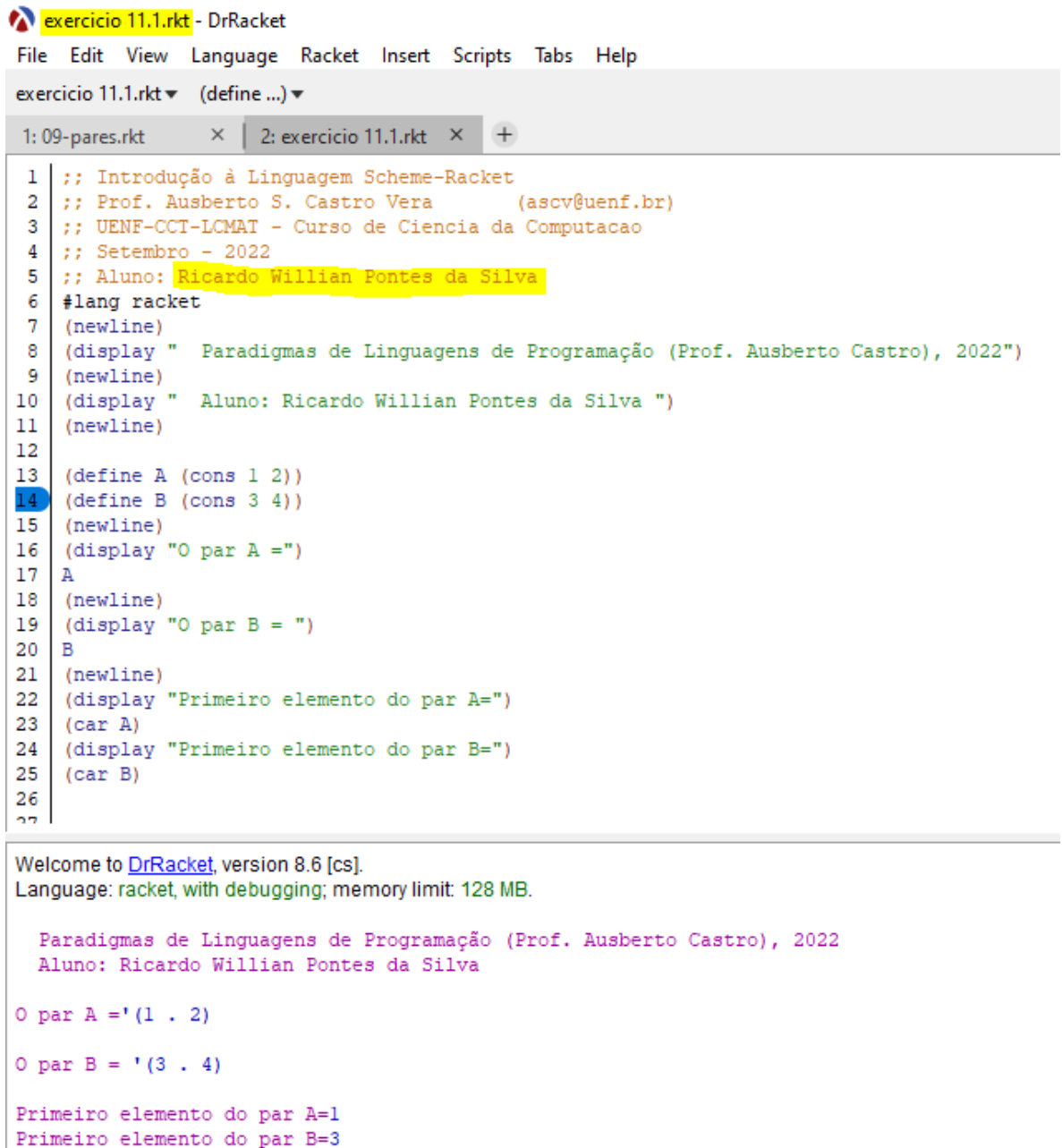
16 ;; PARES CONSTANTES
17 ;; -----
18
19 (define x (cons 1 2))
20 (define y (cons 'a 3))
21 (define z (cons x y)) ; ( (1 2) ('a 3) )
22
23 (define a (cons 1 2))
24 (define b (cons 2 1))
25
26 (define a2 (cons a 3))
27 (define b2 (cons 3 b))
28
29 (define a3 (cons a2 4))
30 (define b3 (cons 4 b2))
31
32 (define a4 (cons a3 5))
33 (define b4 (cons 5 b3))

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022
Aluno: Ricardo Willian Pontes da Silva

O par x = '(1 . 2)
O par y = '(a . 3)
O par z = (x y) = '((1 . 2) a . 3)
O par a4 = '((((1 . 2) . 3) . 4) . 5)
O par b4 = '(5 4 3 2 . 1)
O primeiro elemento do par x = 1
O segundo elemento do par x = 2
>
```

11.1. Escreva um NOVO programa para construir dois pares e indicar em cada um deles o primeiro e o segundo elemento



The screenshot shows the DrRacket IDE with a file named "exercicio 11.1.rkt". The editor contains Scheme code for list manipulation. The code defines two lists, A and B, and then displays their elements and the first element of each. The output window shows the results of these operations.

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera       (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 #lang racket
7 (newline)
8 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022")
9 (newline)
10 (display " Aluno: Ricardo Willian Pontes da Silva ")
11 (newline)
12
13 (define A (cons 1 2))
14 (define B (cons 3 4))
15 (newline)
16 (display "O par A =")
17 A
18 (newline)
19 (display "O par B = ")
20 B
21 (newline)
22 (display "Primeiro elemento do par A=")
23 (car A)
24 (display "Primeiro elemento do par B=")
25 (car B)
26
27
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: `racket`, with debugging; memory limit: 128 MB.

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro), 2022  
Aluno: Ricardo Willian Pontes da Silva

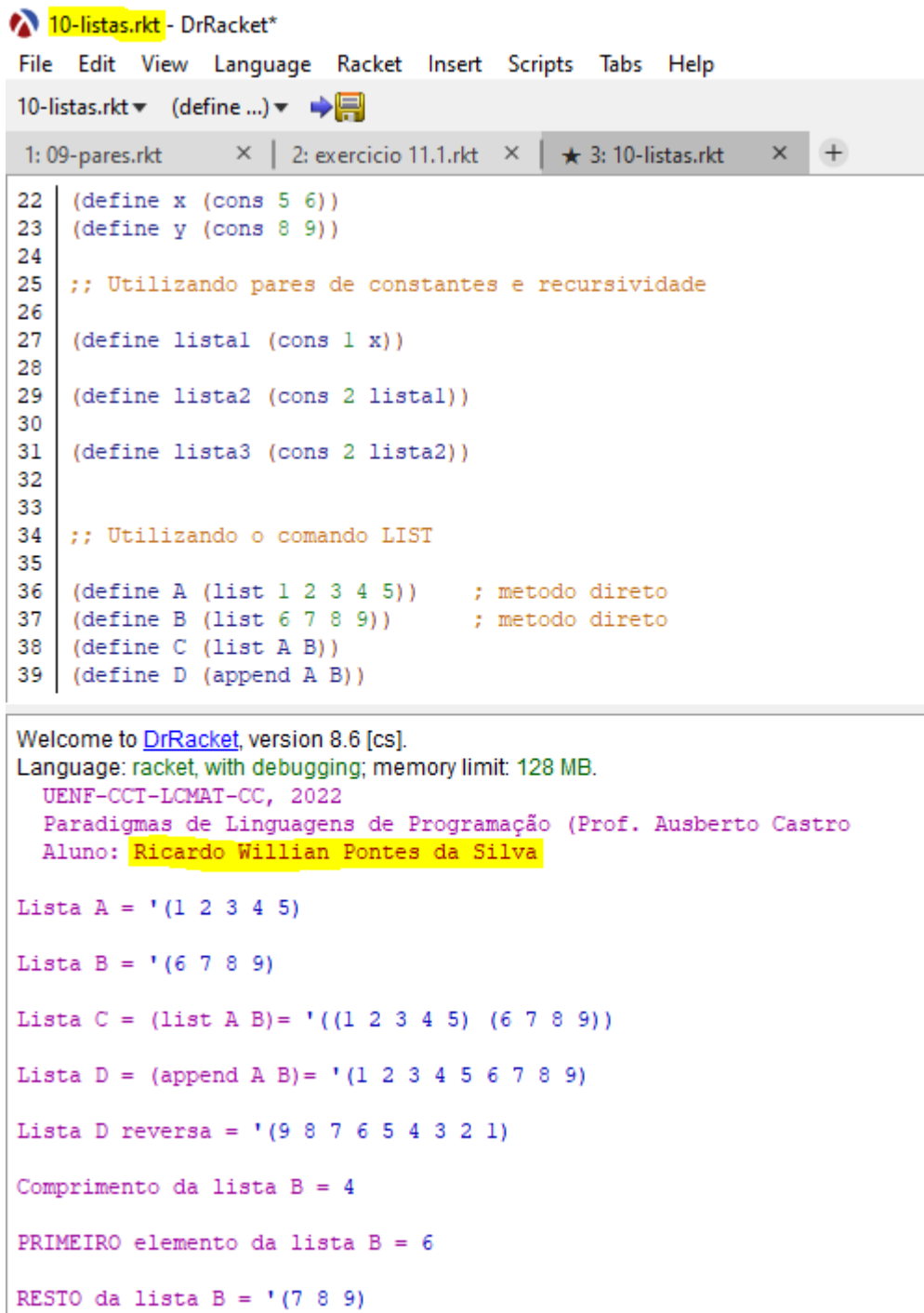
O par A = '(1 . 2)

O par B = '(3 . 4)

Primeiro elemento do par A=1  
Primeiro elemento do par B=3

Arquivo [10-listas.rkt](#) e [11-listas.rkt](#)

12. Executar os programas e observe quantos métodos existem para construir listas

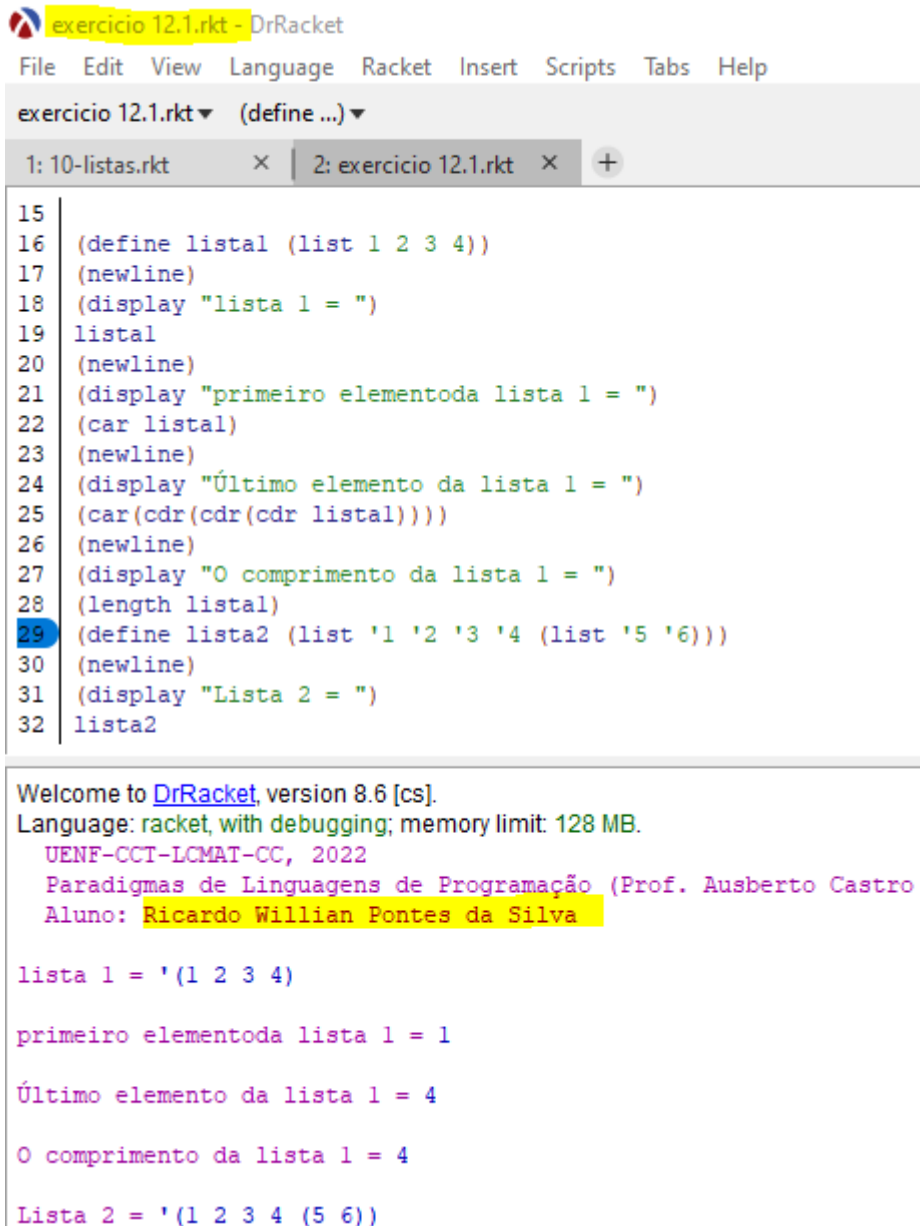


```
10-listas.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
10-listas.rkt (define ...)
1: 09-pares.rkt x 2: exercicio 11.1.rkt x ★ 3: 10-listas.rkt x +
22 (define x (cons 5 6))
23 (define y (cons 8 9))
24
25 ;; Utilizando pares de constantes e recursividade
26
27 (define listal (cons 1 x))
28
29 (define lista2 (cons 2 listal))
30
31 (define lista3 (cons 2 lista2))
32
33
34 ;; Utilizando o comando LIST
35
36 (define A (list 1 2 3 4 5)) ; metodo direto
37 (define B (list 6 7 8 9)) ; metodo direto
38 (define C (list A B))
39 (define D (append A B))

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro
Aluno: Ricardo Willian Pontes da Silva

Lista A = '(1 2 3 4 5)
Lista B = '(6 7 8 9)
Lista C = (list A B)= '((1 2 3 4 5) (6 7 8 9))
Lista D = (append A B)= '(1 2 3 4 5 6 7 8 9)
Lista D reversa = '(9 8 7 6 5 4 3 2 1)
Comprimento da lista B = 4
PRIMEIRO elemento da lista B = 6
RESTO da lista B = '(7 8 9)
```

12.1. Escreva um NOVO programa para construir uma lista e determinar seu primeiro e último elemento, seu comprimento, e uma nova lista com dois elementos a mais que a anterior. Incluir os códigos fonte



The screenshot shows the DrRacket IDE with a file named 'exercicio 12.1.rkt'. The editor contains Racket code that defines a list 'listal' and performs several operations on it, including displaying its elements and length. It also defines a second list 'lista2'. The output window shows the results of these operations: the first list is '(1 2 3 4)', its first element is 1, its last element is 4, and its length is 4. The second list is '(1 2 3 4 (5 6))'.

```
15  
16 (define listal (list 1 2 3 4))  
17 (newline)  
18 (display "lista 1 = ")  
19 listal  
20 (newline)  
21 (display "primeiro elementoda lista 1 = ")  
22 (car listal)  
23 (newline)  
24 (display "Último elemento da lista 1 = ")  
25 (car(cdr(cdr(cdr listal))))  
26 (newline)  
27 (display "O comprimento da lista 1 = ")  
28 (length listal)  
29 (define lista2 (list '1 '2 '3 '4 (list '5 '6)))  
30 (newline)  
31 (display "Lista 2 = ")  
32 lista2
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro  
Aluno: [Ricardo Willian Pontes da Silva](#)

lista 1 = '(1 2 3 4)  
  
primeiro elementoda lista 1 = 1  
  
Último elemento da lista 1 = 4  
  
O comprimento da lista 1 = 4  
  
Lista 2 = '(1 2 3 4 (5 6))

12.2. Utilizando uma ÚNICA linha de comandos, escreva um NOVO programa Racket para construir a lista ( 4 7 2 9 8 7 1 6 2 3 4) a partir das listas A=( 1 2 3 4 ) e B=(5 6 7 8 9)

Arquivo [12-lambda.rtk](#) e [13-lambda.rtk](#)

13. Execute os programas e indique o que faz cada um deles

12-lambda.rkt - DrRacket\*

File Edit View Language Racket Insert Scripts Tabs Help

12-lambda.rkt (define ...) ➡

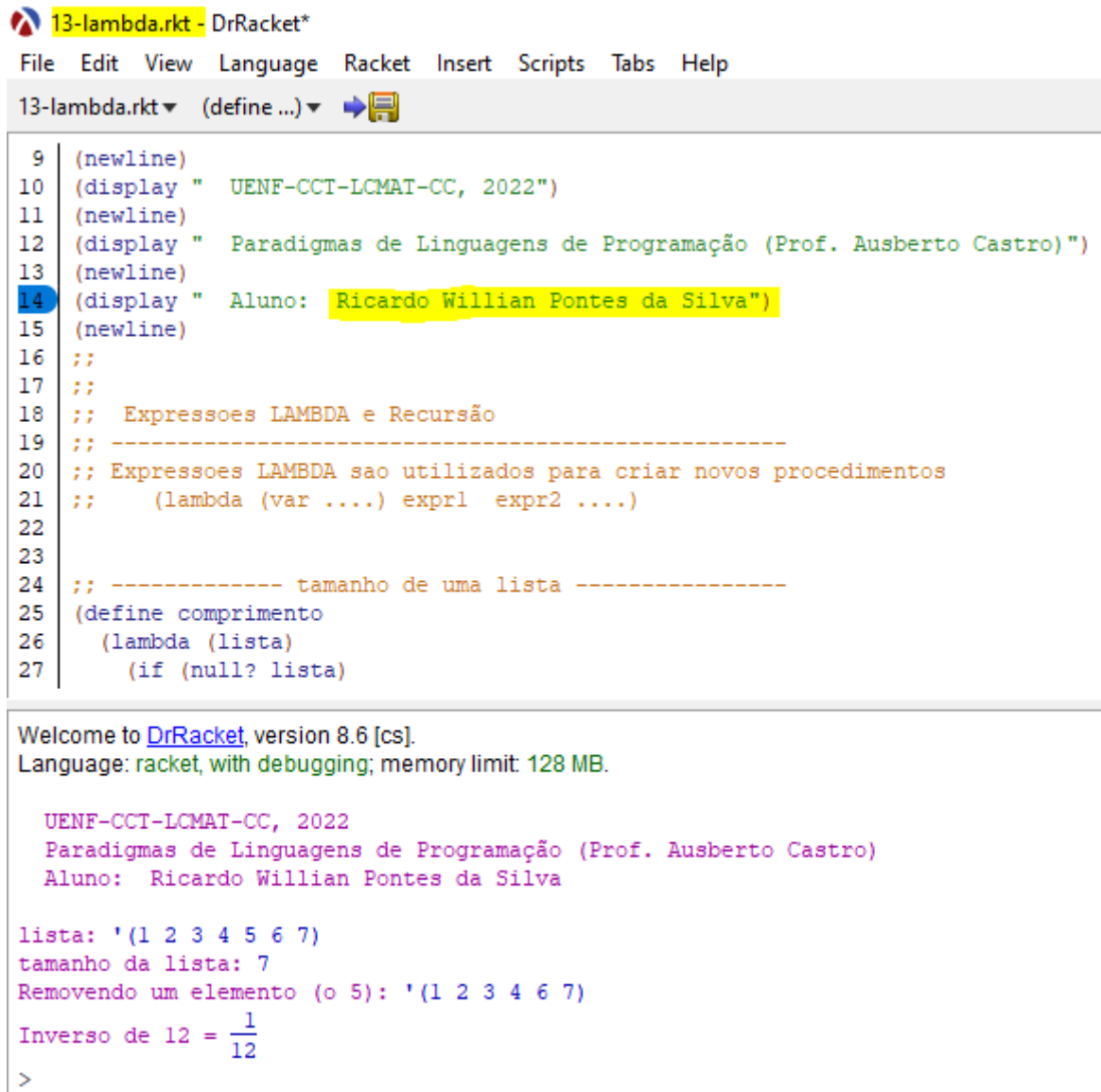
```
7 ; -----
8 (newline)
9 (display " UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display " Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 ;;
16 ;;
17 ;; Expressoes LAMBDA
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.

```
UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Ricardo Willian Pontes da Silva
3+3 = 6
3*3 = 9
3-3 = 0

8+5 = 13
8*5 = 40
8-5 = 3

P(x) = X^2 + 3X - 7    entao P(5) = 33
P(x) = X^2 + 3X - 7    entao P(0) = -7
P(x) = X^2 + 3X - 7    entao P(2) = 3
>
```



```
13-lambda.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
13-lambda.rkt (define ...)
9 (newline)
10 (display " UENF-CCT-LCMAT-CC, 2022")
11 (newline)
12 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display " Aluno: Ricardo Willian Pontes da Silva")
15 (newline)
16 ;;
17 ;;
18 ;; Expressoes LAMBDA e Recursão
19 ;; -----
20 ;; Expressoes LAMBDA sao utilizados para criar novos procedimentos
21 ;; (lambda (var ....) expr1 expr2 ....)
22
23
24 ;; ----- tamanho de uma lista -----
25 (define comprimento
26   (lambda (lista)
27     (if (null? lista)
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: [racket](#), with debugging; memory limit: 128 MB.

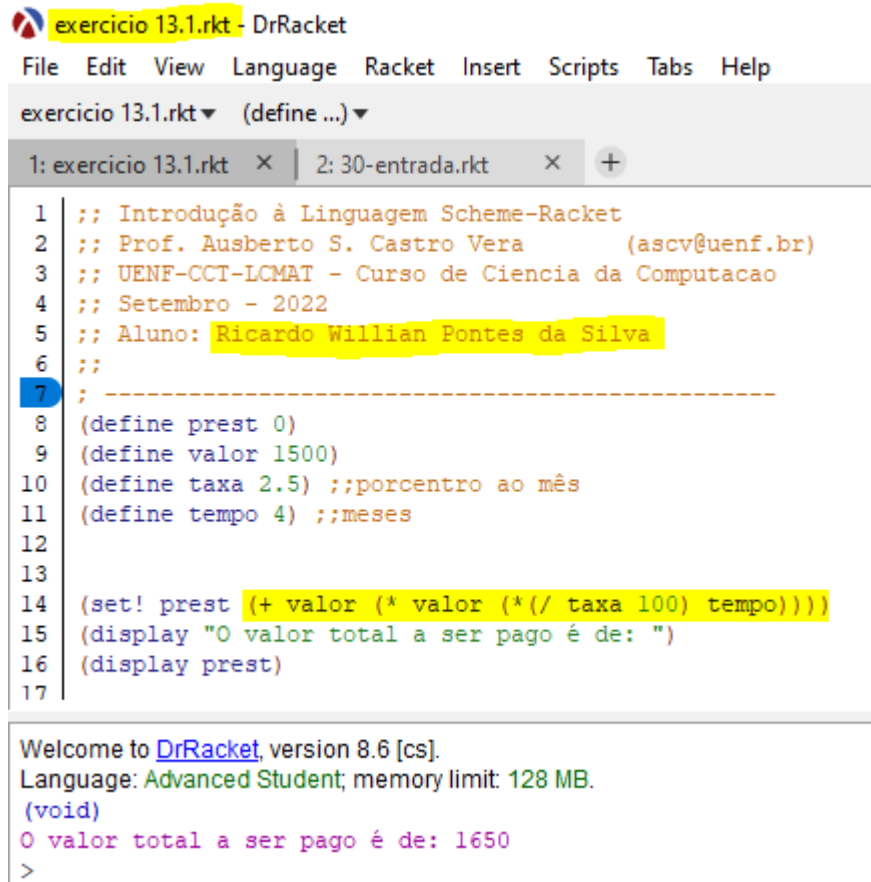
UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: Ricardo Willian Pontes da Silva

lista: '(1 2 3 4 5 6 7)  
tamanho da lista: 7  
Removendo um elemento (o 5): '(1 2 3 4 6 7)  
Inverso de 12 =  $\frac{1}{12}$   
>

O programa 12-lambda tem como objetivo calcular operações matemáticas como soma, subtração, multiplicação e etc, tendo como base um único valor passado como parâmetro, sendo uma forma de enxugar o código e deixa-lo mais simples.

Já o programa 13-lambda visa usar e manipular listas, usando funções para obter informações como seu tamanho, remover um elemento em específico e expressar o inverso de um elemento.

- 13.1. Crie um procedimento para realizar o cálculo de uma prestação em atraso, utilizando a fórmula  $\text{Prest} = \text{valor} + (\text{valor} * (\text{taxa}/100) * \text{tempo})$ . Dar exemplos.



The screenshot shows the DrRacket IDE interface. The title bar reads "exercicio 13.1.rkt - DrRacket". The menu bar includes "File", "Edit", "View", "Language", "Racket", "Insert", "Scripts", "Tabs", and "Help". Below the menu bar, the current file is "exercicio 13.1.rkt" with a dropdown arrow and "(define ...)". The tab bar shows two tabs: "1: exercicio 13.1.rkt" and "2: 30-entrada.rkt". The main editor area contains a Scheme script with the following code:

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 ; -----
8 (define prest 0)
9 (define valor 1500)
10 (define taxa 2.5) ;;porcento ao mês
11 (define tempo 4) ;;meses
12
13
14 (set! prest (+ valor (* valor (/ taxa 100) tempo)))
15 (display "O valor total a ser pago é de: ")
16 (display prest)
17
```


The output area at the bottom shows the following text:

```
Welcome to DrRacket, version 8.6 [cs].
Language: Advanced Student; memory limit: 128 MB.
(void)
O valor total a ser pago é de: 1650
>
```

13.2. O que faz o seguinte procedimento **abcd**:

```
(define abcd
  (lambda (n)
    (let f ((i 2))
      (cond
        ((>= i n) '())
        ((integer? (/ n i))
         (cons i (f (+ i 1))))
        (else (f (+ i 1)))))))
```





 **exercicio 13.2.rkt - DrRacket**  
File Edit View Language Racket Insert Scripts Tabs Help  
exercicio 13.2.rkt ▾ (define ...) ▾  

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (define abcd
10  (lambda (n)
11    (let f ((i 2))
12      (cond
13        ((>= i n) '())
14        ((integer? (/ n i))
15         (cons i (f (+ i 1))))
16        (else (f (+ i 1)))))))
17
18 (display "N=") (abcd 10)
19
```

  
Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
N= (2 5)  
>

O trecho de código acima tem como objetivo expressar através de um parâmetro passado para a função `abcd`, a sua divisão por dois, caso o número passado seja maior ou igual a dois. Com tudo, será impresso também o dividendo da operação.

Arquivo [14-operad-logicos.rtk](#) Operadores lógicos  
14. Executar o programa e indicar o seu conteúdo


 14-operad-logicos.rkt - DrRacket\*  
File Edit View Language Racket Insert Scripts Tabs Help  
14-operad-logicos.rkt (define ...)   

```
9 (newline)
10 (display " UENF-CCT-LCMAT-CC, 2022")
11 (newline)
12 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display " Aluno: Ricardo Willian Pontes da Silva ")
15 (newline)
16 ;;
17 ..
```

  
Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
  
UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: Ricardo Willian Pontes da Silva  
Operadores lógicos...  
#t  
#f  
  
Tabela de valores AND  
T and T = #t  
  
T and F = #f  
  
F and F = #f  
  
Tabela de valores OR  
T or T = #t  
  
T or F = #t  
  
F or F = #f  
  
#t  
#f  
  
not(5 > 3) = #f  
not(10 > 25) = #t  
>

O trecho de código acima tem como objetivo expressar algumas comparações entre os operadores lógicos *and*, que retornará um booleano *true*, quando ambas as condições forem verdadeiras e *or*, que retornará verdadeiro se uma ou mais expressões forem verdade.

14.1. Escreva e teste pelo menos cinco operações lógicas

 **exercicio 14.1.rkt** - DrRacket  
File Edit View Language Racket Insert Scripts Tabs Help  
exercicio 14.1.rkt (define ...)▼  

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9
10 (let ((x 10))
11   (and (> x 5) (< x 11)))
12
13 (let ((y -3))
14   (and (> y 2) (< y 4)))
15
16 (let ((z 7))
17   (and (>= z 7) (> z 4)))
18
19 (let ((a 12))
20   (and (> a a) (> a 4)))
21
22 (let ((b 7))
23   (or (<= b 2) (> b 4)))
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: **racket**, with **debugging**; memory limit: 128 MB.

```
#t
#f
#t
#f
#t
>
```

Arquivo [15-predicados.rtk](#) Predicados

15. Executar o programa e indicar o seu conteúdo

```
15-predicados.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
15-predicados.rkt (define ...) ➡

1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket ;; define a linguagem default
-

car = car? #t
car = cdr? #f
Falso = Falso? #t

Operador EQUIVALENTE
9/2 é EQUIVALENTE a 7/2? #f
12/5 é EQUIVALENTE a 24/10? #t
#t
#t
#t
7 é BOOLEANO? #f
( ) é uma lista NULA? #t
(a) é uma lista NULA? #f
(list 1 3 5 7) é uma lista NULA? #f
#f
Pares?:
Par (3.4) ?: #t
Par 5?: #f
Par (cons 6 9) ?: #t
Numero 6.78 ?: #t
Numero 2/7 ?: #t

É un número INTEIRO 3 ?: #t
É un número INTEIRO 3.0 ?: #t
É un número INTEIRO 3/7 ?: #f

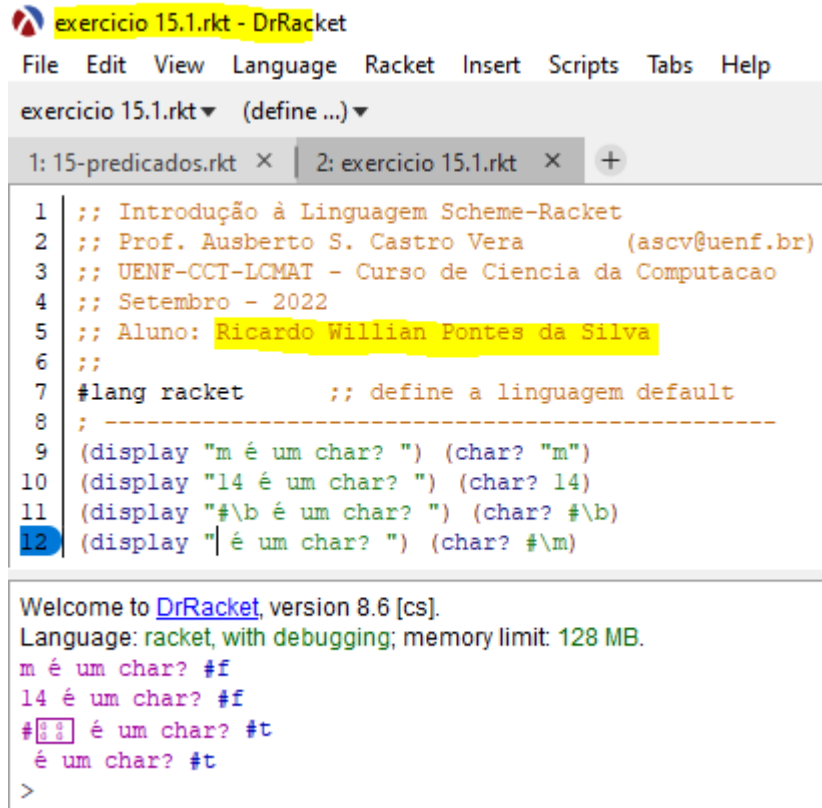
É un número REAL 2 ?: #t
É un número REAL 3/5 ?: #t
É un número REAL 4+5i ?: #f

É un número COMPLEXO 5+3i ?: #t

É un STRING 'Oi, UENF ...' ?: #t
É un STRING 24 ?: #f
>
```

Os predicados retornam um valor booleano após ser utilizado para uma comparação desejada, vale destacar também que tal operador pode ser unido com funções já vistas nos trechos de códigos anteriores como verificar o tipo primitivo de uma variável ou então comparar Strings.

15.1. Testar os predicados: (char? 'm) , (char? 14), (char? #\b) , (char? #\m)



The screenshot shows the DrRacket IDE interface. The title bar reads "exercicio 15.1.rkt - DrRacket". The menu bar includes "File", "Edit", "View", "Language", "Racket", "Insert", "Scripts", "Tabs", and "Help". The toolbar shows "exercicio 15.1.rkt" and "(define ...)". The tab bar displays "1: 15-predicados.rkt" and "2: exercicio 15.1.rkt". The editor window contains the following Racket code:

```
1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (display "m é um char? ") (char? "m")
10 (display "14 é um char? ") (char? 14)
11 (display "#\b é um char? ") (char? #\b)
12 (display "| é um char? ") (char? #\m)
```

The output window shows the following text:

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
m é um char? #f
14 é um char? #f
#\b é um char? #t
| é um char? #t
>
```

Arquivo 16-mapeamentos.rkt Mapeamentos

15.2. Executar o programa e indicar o que ele faz

```
16-mapeamentos.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
16-mapeamentos.rkt (define ...) ➡

10 (display " UENF-CCT-LCMAT-CC, 2022")
11 (newline)
12 (display " Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display " Aluno: Ricardo Willian Pontes da Silva ")
15 (newline)
16 ;;
17 ;;
18 ;; Mapeamentos
19 ; -----
20
21 (define quadrado
22   (lambda ( x ) (* x x)
23   )
24 )
25 ;;;;
26 (define dobro
27   (lambda (y) (* 2 y)
28   )
29 )
30 ;;;;
31 (define proximo
32   (lambda ( x ) (+ x 1)
33   )
34 )

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Fulano
Lista: '(1 4 9 16 25)
Raiz Lista: '(1 2 3 4 5)

Dados: '(2 5 6 23)

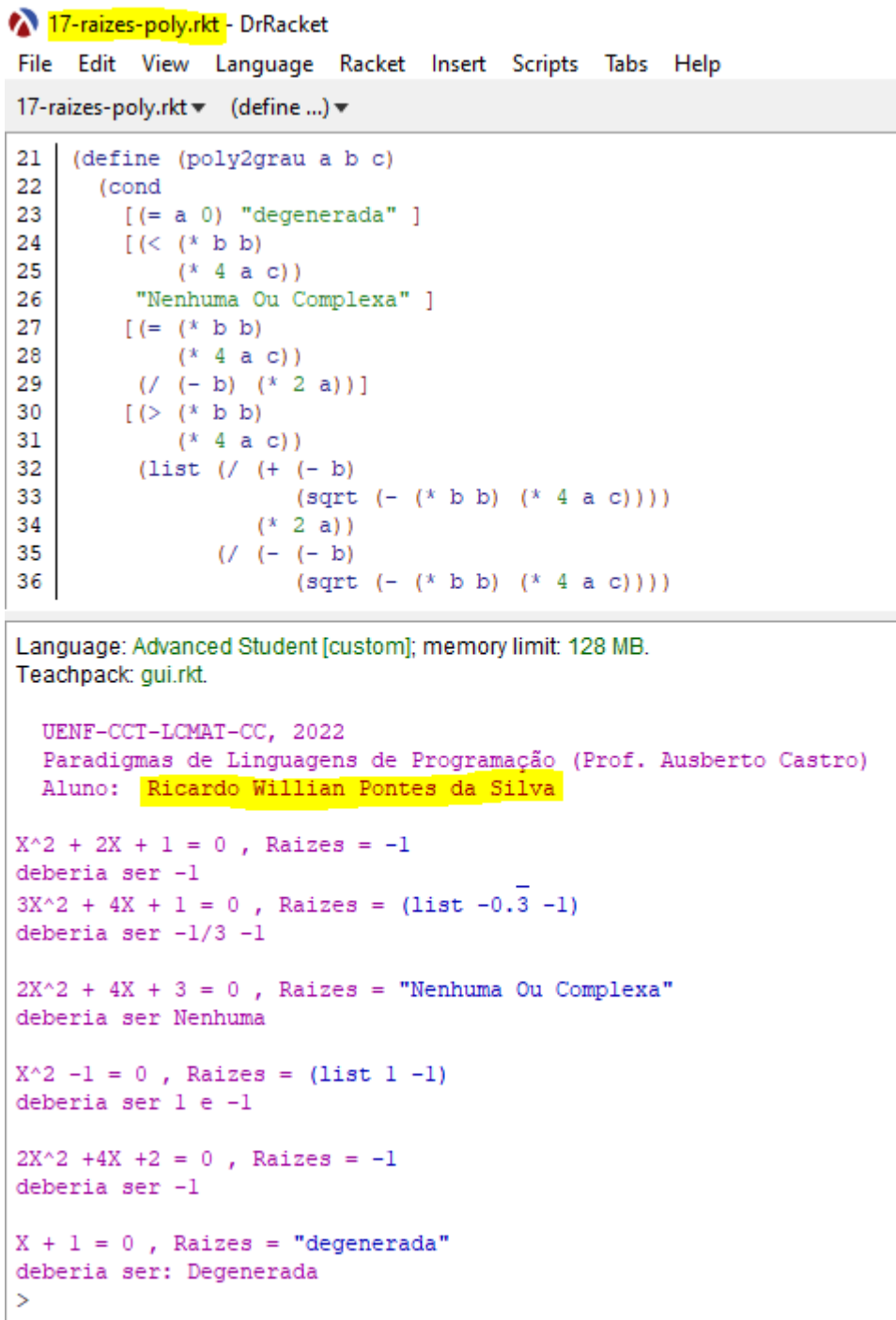
Mapeamentos: Dados-Proximo
'(3 6 7 24)
Mapeamentos: Dados-Quadrado
'(4 25 36 529)
Mapeamentos: Dados-Dobro
'(4 10 12 46)
>
```

A função `map` mostrada no trecho de código acima tem como objetivo percorrer uma determinada lista, tendo como base uma função já pré definida.

- 15.3. Construir um NOVO programa que faça o seguinte mapeamento  
$$x \longrightarrow x^2 + 3x - 9$$

Arquivo [17-raizes-poly.rkt](#) Aplicações: Raízes de polinômios

16. Executar o programa e explicar o conteúdo e os resultados



```
17-raizes-poly.rkt - DrRacket
File Edit View Language Racket Insert Scripts Tabs Help
17-raizes-poly.rkt (define ...)
21 (define (poly2grau a b c)
22   (cond
23     [(= a 0) "degenerada" ]
24     [(< (* b b)
25        (* 4 a c))
26      "Nenhuma Ou Complexa" ]
27     [(= (* b b)
28        (* 4 a c))
29      (/ (- b) (* 2 a))]
30     [(> (* b b)
31        (* 4 a c))
32      (list (/ (+ (- b)
33                 (sqrt (- (* b b) (* 4 a c))))
34             (* 2 a))
35            (/ (- (- b)
36                 (sqrt (- (* b b) (* 4 a c))))
Language: Advanced Student [custom]; memory limit: 128 MB.
Teachpack: gui.rkt.

UNF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Ricardo Willian Pontes da Silva

X^2 + 2X + 1 = 0 , Raizes = -1
deberia ser -1
3X^2 + 4X + 1 = 0 , Raizes = (list -0.3 -1)
deberia ser -1/3 -1

2X^2 + 4X + 3 = 0 , Raizes = "Nenhuma Ou Complexa"
deberia ser Nenhuma

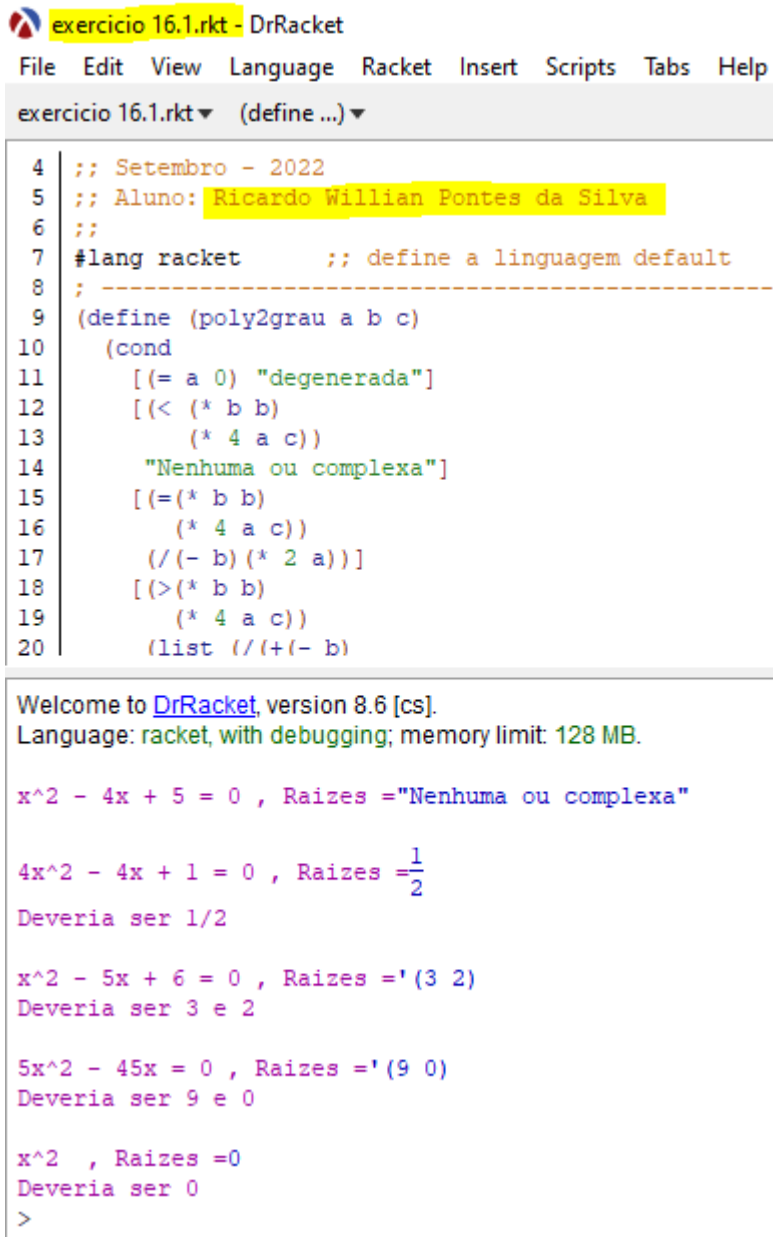
X^2 -1 = 0 , Raizes = (list 1 -1)
deberia ser 1 e -1

2X^2 +4X +2 = 0 , Raizes = -1
deberia ser -1

X + 1 = 0 , Raizes = "degenerada"
deberia ser: Degenerada
>
```

O programa **17-raizes-poly.rtk** tem como objetivo calcular as raízes de polinômios, se é complexa ou degenerada, também expressa qual deveria ser a resposta correta, garantindo seu correto funcionamento.

16.1. Fazer testes para outros cinco polinômios de segundo grau



```
exercício 16.1.rkt - DrRacket
File Edit View Language Racket Insert Scripts Tabs Help
exercício 16.1.rkt (define ...)
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket ;; define a linguagem default
8 ; -----
9 (define (poly2grau a b c)
10   (cond
11     [(= a 0) "degenerada"]
12     [(< (* b b)
13         (* 4 a c))
14      "Nenhuma ou complexa"]
15     [(= (* b b)
16         (* 4 a c))
17      (/ (- b) (* 2 a))]
18     [(> (* b b)
19         (* 4 a c))
20      (list (/ (+ (- b)
21                  (sqrt (- (* b b)
22                          (* 4 a c)))))
23            (/ (- b)
24                (* 2 a)))]))
21
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

x^2 - 4x + 5 = 0 , Raizes ="Nenhuma ou complexa"

4x^2 - 4x + 1 = 0 , Raizes = 1/2
Deveria ser 1/2



x^2 - 5x + 6 = 0 , Raizes ='(3 2)
Deveria ser 3 e 2

5x^2 - 45x = 0 , Raizes ='(9 0)
Deveria ser 9 e 0

x^2 , Raizes =0
Deveria ser 0
>
```

Arquivo **20-estruturas.rtk** Aplicações: Estruturas de dados  
17. Executar o programa e explicar o conteúdo e os resultados



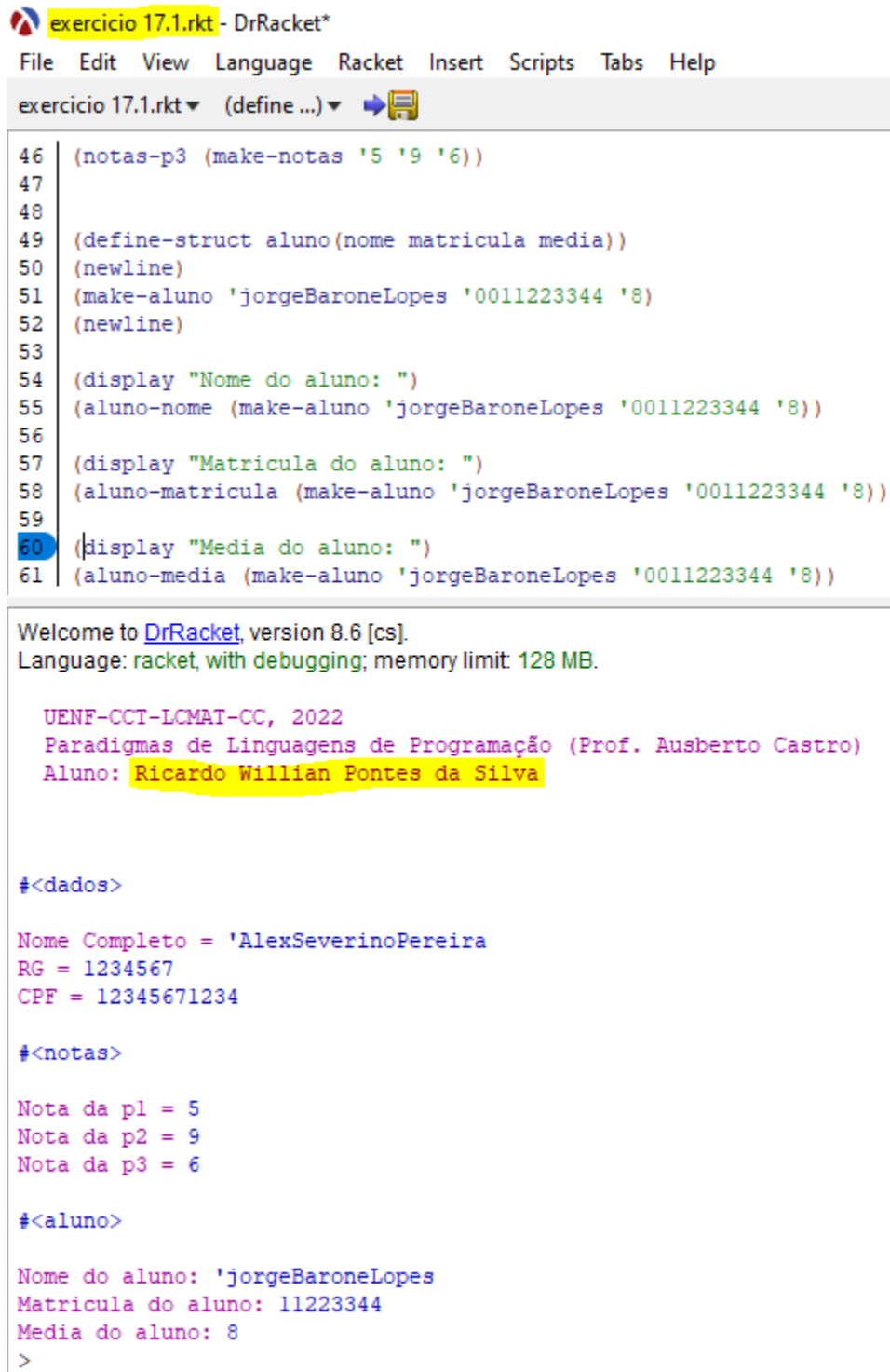
 20-estruturas.rkt - DrRacket\*  
File Edit View Language Racket Insert Scripts Tabs Help  
20-estruturas.rkt (define ...)   

```
21 ;;  
22 (define-struct entrada (nome cep fone))  
23  
24 (newline)  
25 (make-entrada 'PedroSilva '05890-001 '606-7771)  
26 (newline)  
27 (display "Nome = " )  
28 (entrada-nome (make-entrada 'PedroSilva '05890-001 '606-7771))  
29 (display "CEP = " )  
30 (entrada-cep (make-entrada 'PedroSilva '05890-001 '606-7771))  
31 ;;  
32 ;;  
33 (define abc (make-entrada 'PedroSilva '05890-001 '606-7771))  
34 (display "Fone = " )  
35 (entrada-fone abc)  
36
```

  
Welcome to [DrRacket](#), version 8.6 [cs].  
Language: [Advanced Student](#); memory limit: 128 MB.  
  
UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: [Ricardo Willian Pontes da Silva](#)  
  
(make-entrada 'PedroSilva '05890-001 '606-7771)  
  
Nome = 'PedroSilva  
CEP = '05890-001  
Fone = '606-7771  
>

O programa [20-estruturas.rkt](#) utiliza-se da estrutura *struct* para alocar dados de um usuário, como nome, cep e telefone. Logo após, é impresso no dispositivo de saída (monitor) tais informações.

17.1. Em um novo programa defina pelo menos outras TRÊS estruturas diferentes



The screenshot shows the DrRacket IDE interface. The title bar reads "exercicio 17.1.rkt - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu bar is a toolbar with a dropdown menu showing "exercicio 17.1.rkt", a "(define ...)" button, and a "Run" button. The main editor area contains Racket code with line numbers 46 to 61. Line 60 is highlighted. The output area below the editor shows the results of running the code, including a welcome message, language information, and the output of the program's display functions.

```
46 (notas-p3 (make-notas '5 '9 '6))
47
48
49 (define-struct aluno(nome matricula media))
50 (newline)
51 (make-aluno 'jorgeBaroneLopes '0011223344 '8)
52 (newline)
53
54 (display "Nome do aluno: ")
55 (aluno-nome (make-aluno 'jorgeBaroneLopes '0011223344 '8))
56
57 (display "Matricula do aluno: ")
58 (aluno-matricula (make-aluno 'jorgeBaroneLopes '0011223344 '8))
59
60 (display "Media do aluno: ")
61 (aluno-media (make-aluno 'jorgeBaroneLopes '0011223344 '8))
```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: Ricardo Willian Pontes da Silva

#<dados>

Nome Completo = 'AlexSeverinoPereira  
RG = 1234567  
CPF = 12345671234

#<notas>

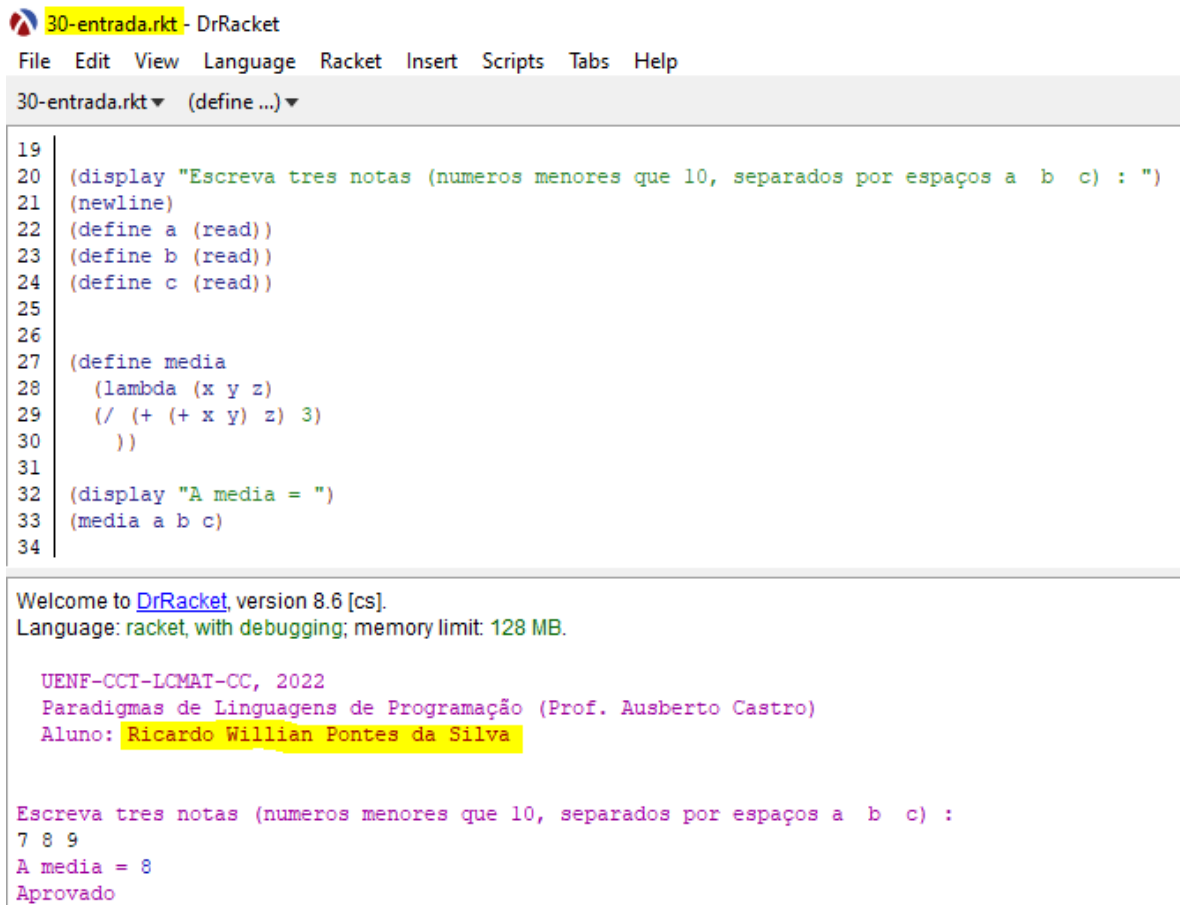
Nota da p1 = 5  
Nota da p2 = 9  
Nota da p3 = 6

#<aluno>

Nome do aluno: 'jorgeBaroneLopes  
Matricula do aluno: 11223344  
Media do aluno: 8  
>

Arquivo [30-entrada.rtk](#)

18. Executar o programa e explicar o conteúdo e os resultados



```
30-entrada.rkt - DrRacket
File Edit View Language Racket Insert Scripts Tabs Help
30-entrada.rkt (define ...)
19
20 (display "Escreva tres notas (numeros menores que 10, separados por espaços a b c) : ")
21 (newline)
22 (define a (read))
23 (define b (read))
24 (define c (read))
25
26
27 (define media
28   (lambda (x y z)
29     (/ (+ (+ x y) z) 3)
30   ))
31
32 (display "A media = ")
33 (media a b c)
34

Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2022
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
Aluno: Ricardo Willian Pontes da Silva

Escreva tres notas (numeros menores que 10, separados por espaços a b c) :
7 8 9
A media = 8
Aprovado
```

O programa **30-entrada.rkt** utiliza-se do recurso de entrada de dados através do teclado para armazenar valores e retornar a média aritmética entre os dígitos armazenados.

- 18.1. Criar um programa NOVO que faça a leitura de dados pessoais de duas pessoas (utilize entrada de dados)

```

20 (display "informe os dados pessoais da primeira pessoa(Nome , RG , CPF) :")
21 (newline)
22 (define N1 (read))
23 (define R1 (read))
24 (define C1 (read))
25
26 (display "informe os dados pessoais da segunda pessoa(Nome , RG , CPF) :")
27 (newline)
28 (define N2 (read))
29 (define R2 (read))
30 (define C2 (read))
31
32 (display "Nome da primeira pessoa: ")
33 N1
34
35 (display "CPF da primeira pessoa: ")

```

Welcome to [DrRacket](#), version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.

UENF-CCT-LCMAT-CC, 2022  
Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)  
Aluno: Ricardo Willian Pontes da Silva

```

informe os dados pessoais da primeira pessoa(Nome , RG , CPF) :
Ricardo
1111111
2222222
informe os dados pessoais da segunda pessoa(Nome , RG , CPF) :
Willian
333333
5555555
Nome da primeira pessoa: 'Ricardo
CPF da primeira pessoa: 2222222
RG da primeira pessoa: 1111111
Nome da segunda pessoa: 'Willian
CPF da segunda pessoa: 5555555
RG da segunda pessoa: 333333

```

## Parte 2:

Resolver a lista de Exercícios no final da Notas de Aula (Slide 50)

```

1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display "  UENF-CCT-LCMAT-CC, 2022")
11 (newline)
12 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno: Ricardo Willian Pontes da Silva ")
15 (newline)
16 ;;
17
18 (define fun
19   (lambda (x y)
20     (-(/(+ 2 x) (- (* 3 y) 4)) (+ (* x y) 5)))
21   |
22   )
23 (fun 2 1)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

```

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Ricardo Willian Pontes da Silva
-11
>

```

```

2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (newline)
10 (display "  UENF-CCT-LCMAT-CC, 2022")
11 (newline)
12 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
13 (newline)
14 (display "  Aluno: Ricardo Willian Pontes da Silva ")
15 (newline)
16 ;;
17
18 (define fun
19   (lambda (x)
20     (- (+ (* x x) (* 3 x)) 5)
21   )
22 )
23
24 (fun 2)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

```

  UENF-CCT-LCMAT-CC, 2022
  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)
  Aluno: Ricardo Willian Pontes da Silva

```

5  
>

```

1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket          ;; define a linguagem default
8 ; -----
9 (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 (define listal (list 3 5 7 10 2))
16 (newline)
17 (display "Lista 1 = ")
18 listal
19 (newline)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Lista 1 = '(3 5 7 10 2)

```

1  ;; Introdução à Linguagem Scheme-Racket
2  ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3  ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4  ;; Setembro - 2022
5  ;; Aluno: Ricardo Willian Pontes da Silva
6  ;;
7  #lang racket          ;; define a linguagem default
8  ; -----
9  (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 (define listal (list 3 5 7 10 2))
16 (newline)
17 (display "Lista 1 = ")
18 listal
19 (newline)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Lista 1 = '(3 5 7 10 2)

Segundo elemento : 5



```

1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket          ;; define a linguagem default
8 ; -----
9 (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 (define listal (list 3 5 7 10 2))
16 (newline)
17 (display "Lista 1 = ")
18 listal
19 (newline)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Lista 1 = '(3 5 7 10 2)

Segundo elemento : 5

Antepenúltimo elemento : 7

```

1 ;; Introdução à Linguagem Scheme-Racket
2 ;; Prof. Ausberto S. Castro Vera      (ascv@uenf.br)
3 ;; UENF-CCT-LCMAT - Curso de Ciencia da Computacao
4 ;; Setembro - 2022
5 ;; Aluno: Ricardo Willian Pontes da Silva
6 ;;
7 #lang racket      ;; define a linguagem default
8 ; -----
9 (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 (define listal (list 3 5 7 10 2))
16 (newline)
17 (display "Lista 1 = ")
18 listal
19 (newline)
20
21 (display "Digite um elemento a ser consultado: ")
22 (define a(read))
23
24 (display "O elemento pertence a lista 1? ")
25 (member a (list 1 2 3 4 5))
26 (newline)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Lista 1 = '(3 5 7 10 2)

Digite um elemento a ser consultado: 4

O elemento 4 pertence a lista 1? '(4 5)

```

8  ; -----
9  (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15 (define lista1 (list 3 5 7 10 2))
16 (newline)
17 (display "Lista 1 = ")
18 lista1
19 (newline)
20 |
21 (define lista2 (list 'a 'b 'c 'd 'e))
22 (display "Lista 2 = ")
23 lista2
24 (newline)
25
26 (define lista2a (cons 'a (cons 'b (cons 'c (cons 'd (cons 'e '(3)))))))
27 (display "Lista 2 com terceiro elemento no final da lista: ")
28 lista2a
29 (newline)
30
31 (define lista2b (cons '3 '(a b c d e)))
32 (display "Lista 2 com terceiro elemento no inicio da lista: ")
33 lista2b

```

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Lista 1 = '(3 5 7 10 2)

Lista 2 = '(a b c d e)

Lista 2 com terceiro elemento no final da lista: '(a b c d e 3)

Lista 2 com terceiro elemento no inicio da lista: '(3 a b c d e)

```

6  ;;
7  #lang racket      ;; define a linguagem default
8  ; -----
9  (display "  UENF-CCT-LCMAT-CC, 2022")
10 (newline)
11 (display "  Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)")
12 (newline)
13 (display "  Aluno: Ricardo Willian Pontes da Silva ")
14 (newline)
15
16 (define (Pquadrado L)
17   (* L 4))
18 (display "Perímetro do quadrado de lado 5: ")
19 (Pquadrado 5)
20 (newline)
21
22 (define (Pcirculo R)
23   ((* 2 3.14)R ))
24 (display "Perímetro do circulo de raio 9: ")
25 (Pcirculo 9)
26 (newline)
27
28 (define (Ptriangulo A B C)
29   (+ (+ A B)C ))
30 (display "Perímetro do triangulo de lado A = 3 B = 9 C = 3 ")
31 (Ptriangulo 3 9 3)

```

Welcome to [DrRacket](#), version 8.6 [cs].

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

UENF-CCT-LCMAT-CC, 2022

Paradigmas de Linguagens de Programação (Prof. Ausberto Castro)

Aluno: Ricardo Willian Pontes da Silva

Perímetro do quadrado de lado 5: 20

Perímetro do circulo de raio 9: 56.52

Perímetro do triangulo de lado A = 3 B = 9 C = 3 15