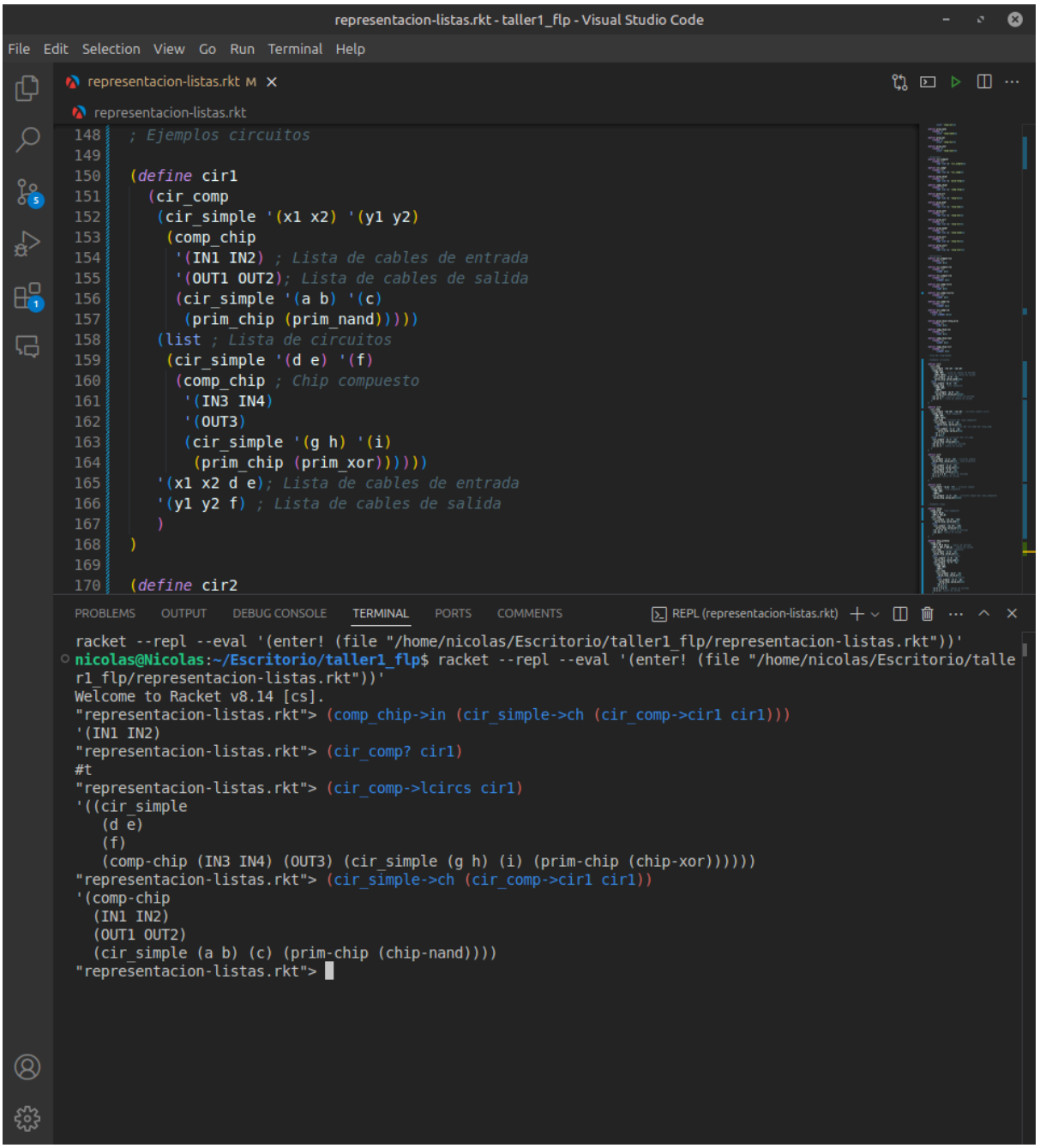
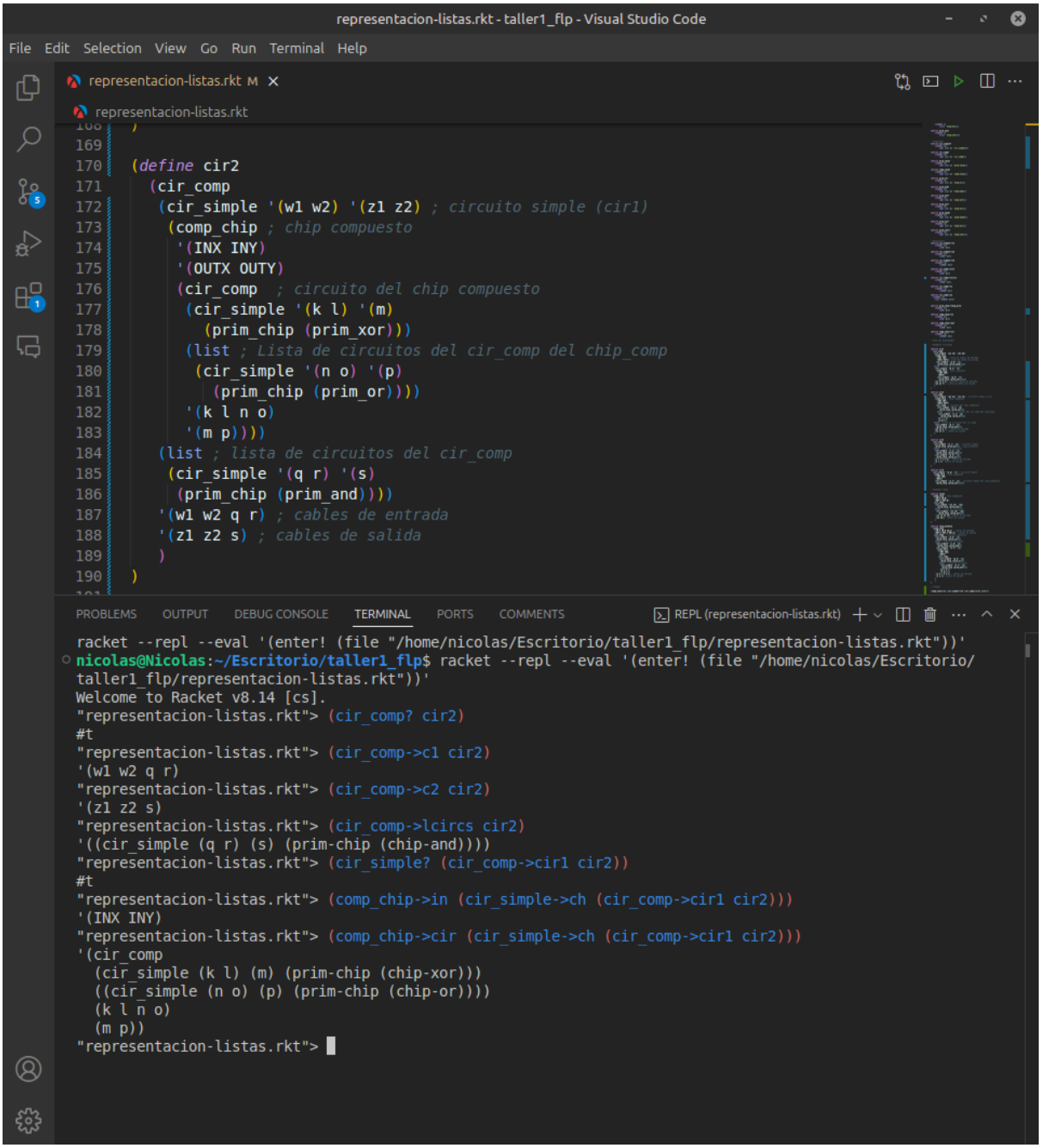
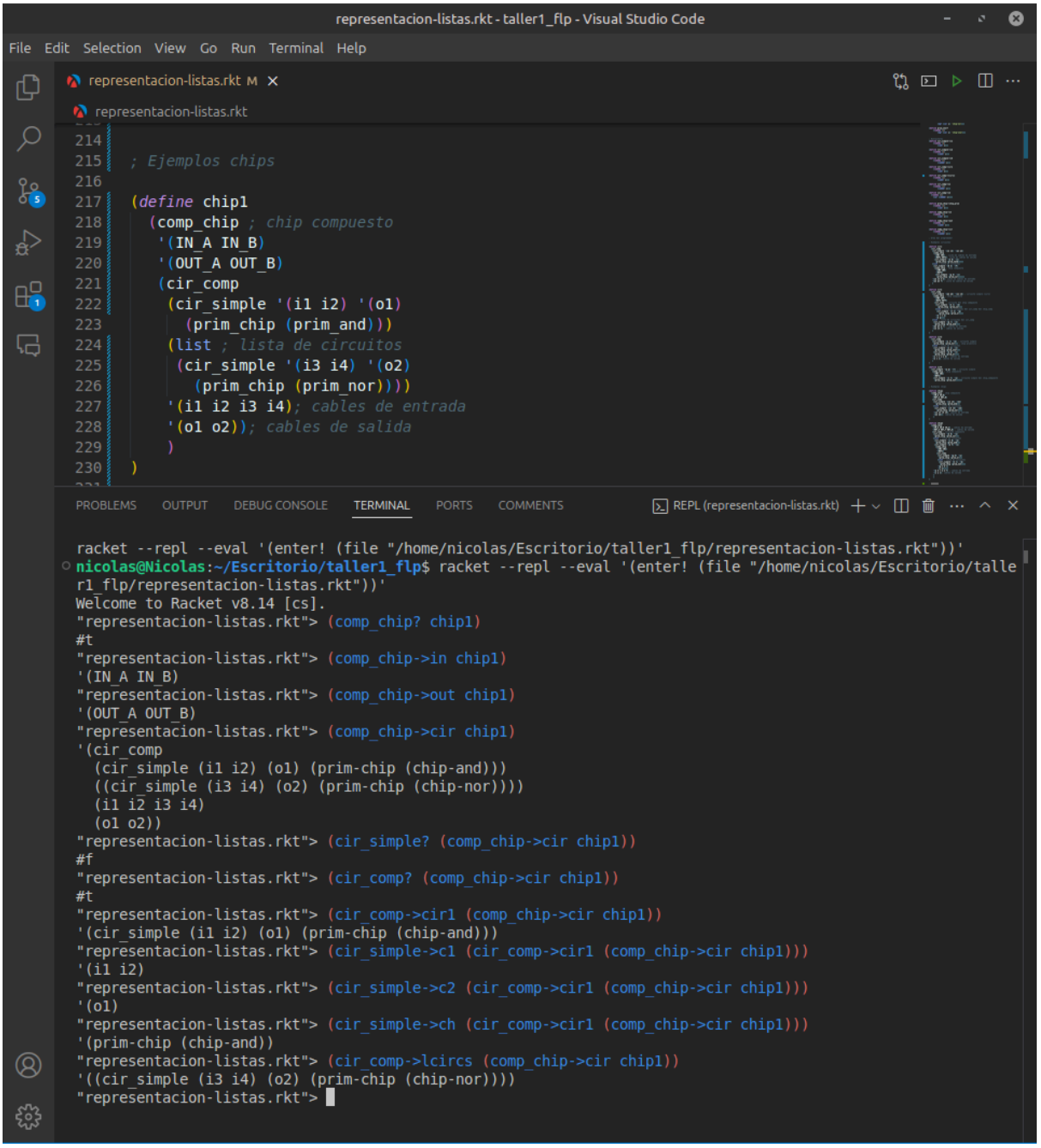
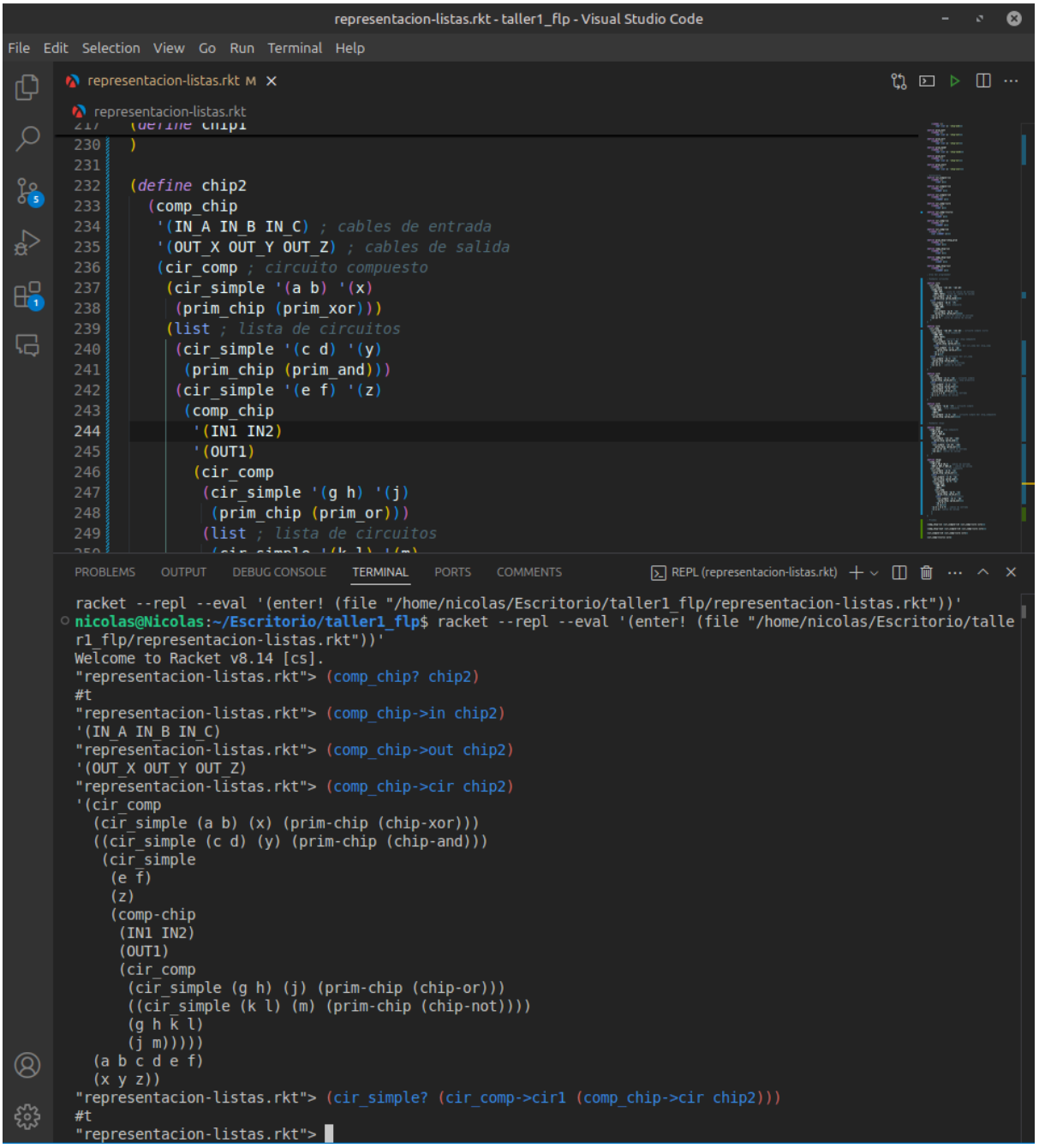
Integrantes:
Anderson Gomez -2266242
Nicolas Rodriguez - 2266071
Michael Rodriguez -2266193
Jhon Alexis Ruiz - 2266014

CAPTURAS REPRESENTACIÓN - LISTAS:




representacion-listas.rkt - taller1_flp - Visual Studio Code
File Edit Selection View Go Run Terminal Help
representacion-listas.rkt M
representacion-listas.rkt
)
(define cir2
(cir_comp
(cir_simple '(w1 w2) '(z1 z2) ; circuito simple (cir1)
(comp_chip ; chip compuesto
'(INX INY)
'(OUTX OUTY)
(cir_comp ; circuito del chip compuesto
(cir_simple '(k l) '(m)
(prim_chip (prim_xor)))
(list ; Lista de circuitos del cir_comp del chip_comp
(cir_simple '(n o) '(p)
(prim_chip (prim_or))))
'(k l n o)
'(m p))))
(list ; lista de circuitos del cir_comp
(cir_simple '(q r) '(s)
(prim_chip (prim_and))))
'(w1 w2 q r) ; cables de entrada
'(z1 z2 s) ; cables de salida
)
)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
REPL (representacion-listas.rkt)
racket --repl --eval '(enter! (file "/home/nicolas/Escritorio/taller1_flp/representacion-listas.rkt"))'
nicolas@Nicolas:~/Escritorio/taller1_flp$ racket --repl --eval '(enter! (file "/home/nicolas/Escritorio/
taller1_flp/representacion-listas.rkt"))'
Welcome to Racket v8.14 [cs].
"representacion-listas.rkt"> (cir_comp? cir2)
#t
"representacion-listas.rkt"> (cir_comp->c1 cir2)
'(w1 w2 q r)
"representacion-listas.rkt"> (cir_comp->c2 cir2)
'(z1 z2 s)
"representacion-listas.rkt"> (cir_comp->lcircs cir2)
'((cir_simple (q r) (s) (prim-chip (chip-and))))
"representacion-listas.rkt"> (cir_simple? (cir_comp->cir1 cir2))
#t
"representacion-listas.rkt"> (comp_chip->in (cir_simple->ch (cir_comp->cir1 cir2)))
'(INX INY)
"representacion-listas.rkt"> (comp_chip->cir (cir_simple->ch (cir_comp->cir1 cir2)))
'(cir_comp
(cir_simple (k l) (m) (prim-chip (chip-xor)))
((cir_simple (n o) (p) (prim-chip (chip-or))))
(k l n o)
(m p))
"representacion-listas.rkt">
Ln 190, Col 2 Spaces: 2 UTF-8 LF Racket


representacion-listas.rkt - taller1_flp - Visual Studio Code
File Edit Selection View Go Run Terminal Help
representacion-listas.rkt M
representacion-listas.rkt
214
215 ; Ejemplos chips
216
217 (define chip1
218 (comp_chip ; chip compuesto
219 '(IN_A IN_B)
220 '(OUT_A OUT_B)
221 (cir_comp
222 (cir_simple '(i1 i2) '(o1)
223 (prim_chip (prim_and)))
224 (list ; lista de circuitos
225 (cir_simple '(i3 i4) '(o2)
226 (prim_chip (prim_nor))))
227 '(i1 i2 i3 i4); cables de entrada
228 '(o1 o2)); cables de salida
229)
230)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
REPL (representacion-listas.rkt)
racket --repl --eval '(enter! (file "/home/nicolas/Escritorio/taller1_flp/representacion-listas.rkt"))'
nicolas@Nicolas:~/Escritorio/taller1_flp$ racket --repl --eval '(enter! (file "/home/nicolas/Escritorio/taller1_flp/representacion-listas.rkt"))'
Welcome to Racket v8.14 [cs].
"representacion-listas.rkt"> (comp_chip? chip1)
#t
"representacion-listas.rkt"> (comp_chip->in chip1)
'(IN_A IN_B)
"representacion-listas.rkt"> (comp_chip->out chip1)
'(OUT_A OUT_B)
"representacion-listas.rkt"> (comp_chip->cir chip1)
'(cir_comp
(cir_simple (i1 i2) (o1) (prim-chip (chip-and)))
((cir_simple (i3 i4) (o2) (prim-chip (chip-nor))))
(i1 i2 i3 i4)
(o1 o2))
"representacion-listas.rkt"> (cir_simple? (comp_chip->cir chip1))
#f
"representacion-listas.rkt"> (cir_comp? (comp_chip->cir chip1))
#t
"representacion-listas.rkt"> (cir_comp->cir1 (comp_chip->cir chip1))
'(cir_simple (i1 i2) (o1) (prim-chip (chip-and)))
"representacion-listas.rkt"> (cir_simple->c1 (cir_comp->cir1 (comp_chip->cir chip1)))
'(i1 i2)
"representacion-listas.rkt"> (cir_simple->c2 (cir_comp->cir1 (comp_chip->cir chip1)))
'(o1)
"representacion-listas.rkt"> (cir_simple->ch (cir_comp->cir1 (comp_chip->cir chip1)))
'(prim-chip (chip-and))
"representacion-listas.rkt"> (cir_comp->lcircs (comp_chip->cir chip1))
'((cir_simple (i3 i4) (o2) (prim-chip (chip-nor))))
"representacion-listas.rkt">



EJEMPLOS REPRESENTACION PROCEDIMIENTOS:

```
365
366  (define circuito5
367    (cir_comp
368      '(a b c d z)
369      '(aa)
370      (cir_comp
371        '(a b c d)
372        '(w x y)
373        (cir_simple '(a b) '(w)
374          (prim_chip (prim_nor)))
375        (list
376          (cir_simple '(c d) '(x)
377            (prim_chip (prim_or)))
378          (cir_simple '(w x) '(y)
379            (prim_chip (prim_and)))))
380      (list
381        (cir_simple '(y z) '(aa)
382          (prim_chip (prim_xor))))))
383
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS REP

```
                                                    circuito5)
"taller1_flp\\representacion-procedimientos.rkt"> (cir_simple?
                                                    circuito5)
#f
"taller1_flp\\representacion-procedimientos.rkt"> (cir_comp?
                                                    circuito5)
#t
"taller1_flp\\representacion-procedimientos.rkt"> (circuito5 1)
'(a b c d z)
"taller1_flp\\representacion-procedimientos.rkt"> (circuito5 2)
'(aa)
"taller1_flp\\representacion-procedimientos.rkt"> (circuito5 3)
#<procedure:...-procedimientos.rkt:16:8>
"taller1_flp\\representacion-procedimientos.rkt"> 
```

```
383
384 (define circuito6
385   (cir_comp
386     '(a b c d ee)
387     '(ff)
388     (cir_comp
389       '(a b c d)
390       '(bb cc dd)
391       (cir_simple '(a b) '(bb)
392         (prim_chip (prim_and)))
393       (list
394         (cir_simple '(c d) '(cc)
395           (prim_chip (prim_xor)))
396         (cir_simple '(bb cc) '(dd)
397           (prim_chip (prim_nor)))))
398     (list
399       (cir_simple '(dd ee) '(ff)
400         (prim_chip (prim_nand))))))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS REPL (representa

```
PS C:\Users\Ander\Desktop\FLP\Flp_taller 1> racket --repl --eval '(enter! (file \"
er 1/taller1_flp/representacion-procedimientos.rkt\"))'
Welcome to Racket v8.14 [cs].
"taller1_flp\\representacion-procedimientos.rkt"> (circuito6 0)
'(a b c d ee)
"taller1_flp\\representacion-procedimientos.rkt"> (circuito6 2)
'(ff)
"taller1_flp\\representacion-procedimientos.rkt"> (circuito6 3)
#<procedure:...-procedimientos.rkt:16:8>
```

```
(define chip6
  (comp_chip
  '()
  '(OUTA)
  (cir_comp
    '(a b c d ee)
    '(ff)
    (cir_comp
      '(a b c d)
      '(bb cc dd)
      (cir_simple '(a b) '(bb)
        (prim_chip (prim_and)))
      (list
        (cir_simple '(c d) '(cc)
          (prim_chip (prim_xor)))
        (cir_simple '(bb cc) '(dd)
          (prim_chip (prim_nor)))))
    (list
      (cir_simple '(dd ee) '(ff)
        (prim_chip (prim_nand)))))))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS REPL (representacion-pr

```
PS C:\Users\Ander\Desktop\FLP\Flp_taller 1> racket --repl --eval '(enter! (file \"c:/Us
er 1/taller1_flp/representacion-procedimientos.rkt\"))'
Welcome to Racket v8.14 [cs].
"taller1_flp\\representacion-procedimientos.rkt"> (chip1 0)
"taller1_flp\\representacion-procedimientos.rkt"> (chip1 1)
'(INA INB INC IND)
"taller1_flp\\representacion-procedimientos.rkt"> (chip1 2)
'(OUTA)
"taller1_flp\\representacion-procedimientos.rkt"> (chip1 3)
#<procedure:...-procedimientos.rkt:16:8>
```

```
(define chip3
  (comp_chip
  '(INA INB INC)
  '(OUTA)
  (cir_comp
    '(a b c d p)
    '(q r)
    (cir_comp
      '(a b c d)
      '(m n o)
      (cir_simple '(a b) '(m)
        (prim_chip (prim_nand)))
      (list
        (cir_simple '(c d) '(n)
          (prim_chip (prim_nor)))
        (cir_simple '(m n) '(o)
          (prim_chip (prim_xor)))))
    (list
      (cir_simple '(o p) '(q)
        (prim_chip (prim_and)))))))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS REPL (representa

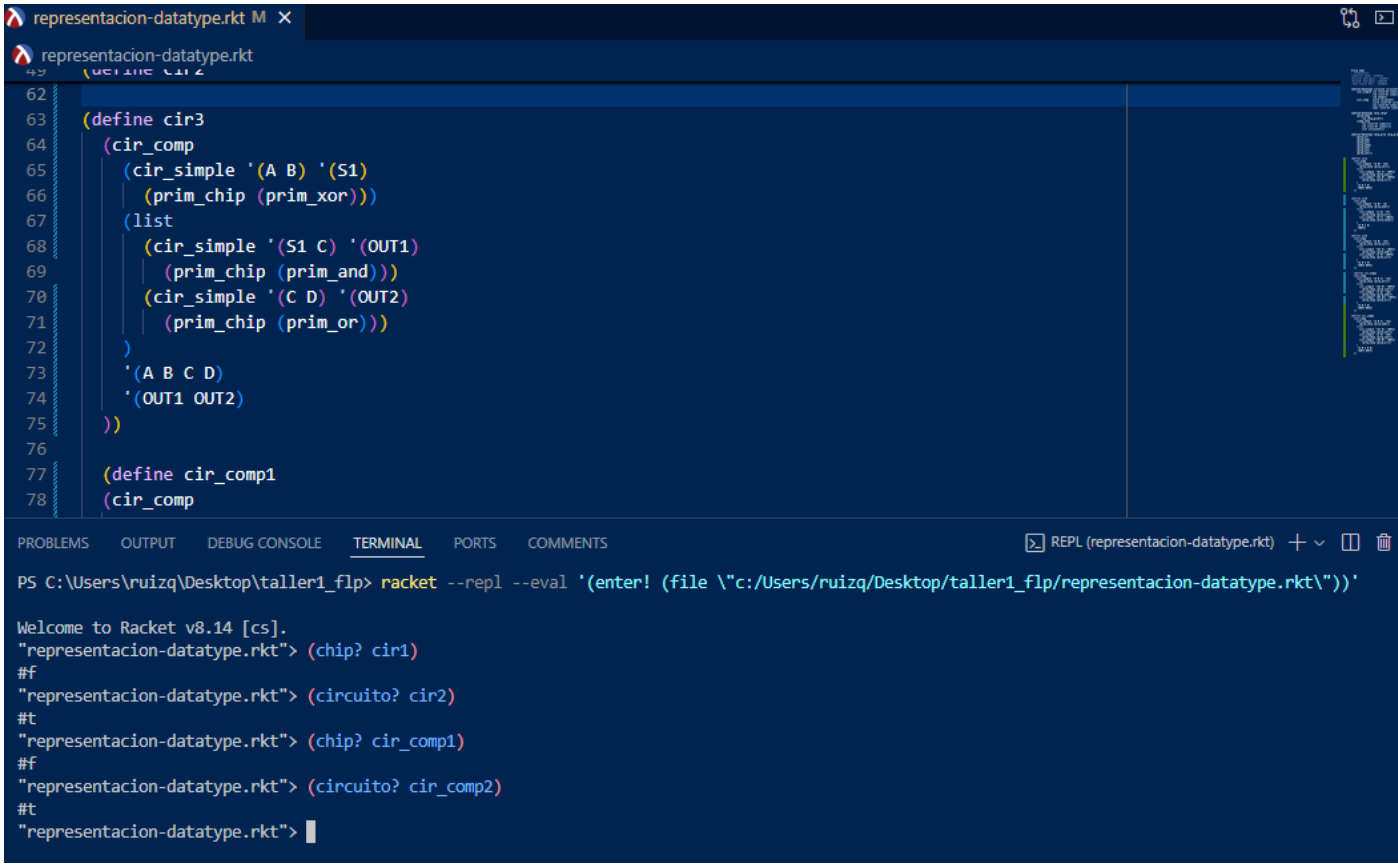
```
er 1/taller1_flp/representacion-procedimientos.rkt\"))'
Welcome to Racket v8.14 [cs].
"taller1_flp\\representacion-procedimientos.rkt"> (comp_chip? chip3)
#t
"taller1_flp\\representacion-procedimientos.rkt"> (comp_chip->in chip3)
'(INA INB INC)
"taller1_flp\\representacion-procedimientos.rkt"> (chip3 1)
'(INA INB INC)
"taller1_flp\\representacion-procedimientos.rkt"> (comp_chip->out chip3)
'(OUTA)
"taller1_flp\\representacion-procedimientos.rkt"> (chip3 2)
'(OUTA)
"taller1_flp\\representacion-procedimientos.rkt"> (chip3 3)
#<procedure:...-procedimientos.rkt:16:8>
```

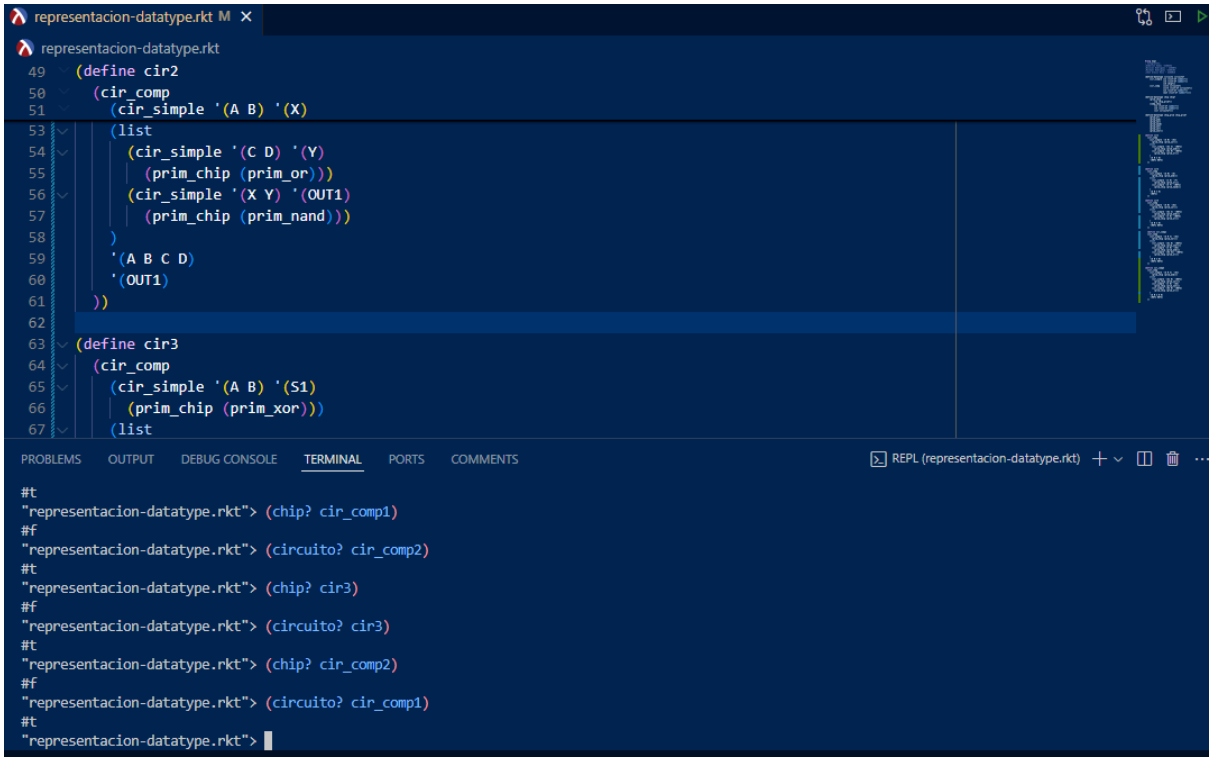
```
(define chip2
  (comp_chip
  '(INA INB)
  '(OUTA OUTB)
  (cir_comp
    '(a b c d k)
    '(l m)
    (cir_comp
      '(a b c d)
      '(h i j)
      (cir_simple '(a b) '(h)
        (prim_chip (prim_or)))
      (list
        (cir_simple '(c d) '(i)
          (prim_chip (prim_xor)))
        (cir_simple '(h i) '(j)
          (prim_chip (prim_and)))))
    (list
      (cir_simple '(j k) '(l)
        (prim_chip (prim_nand)))))))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS REPL (rep

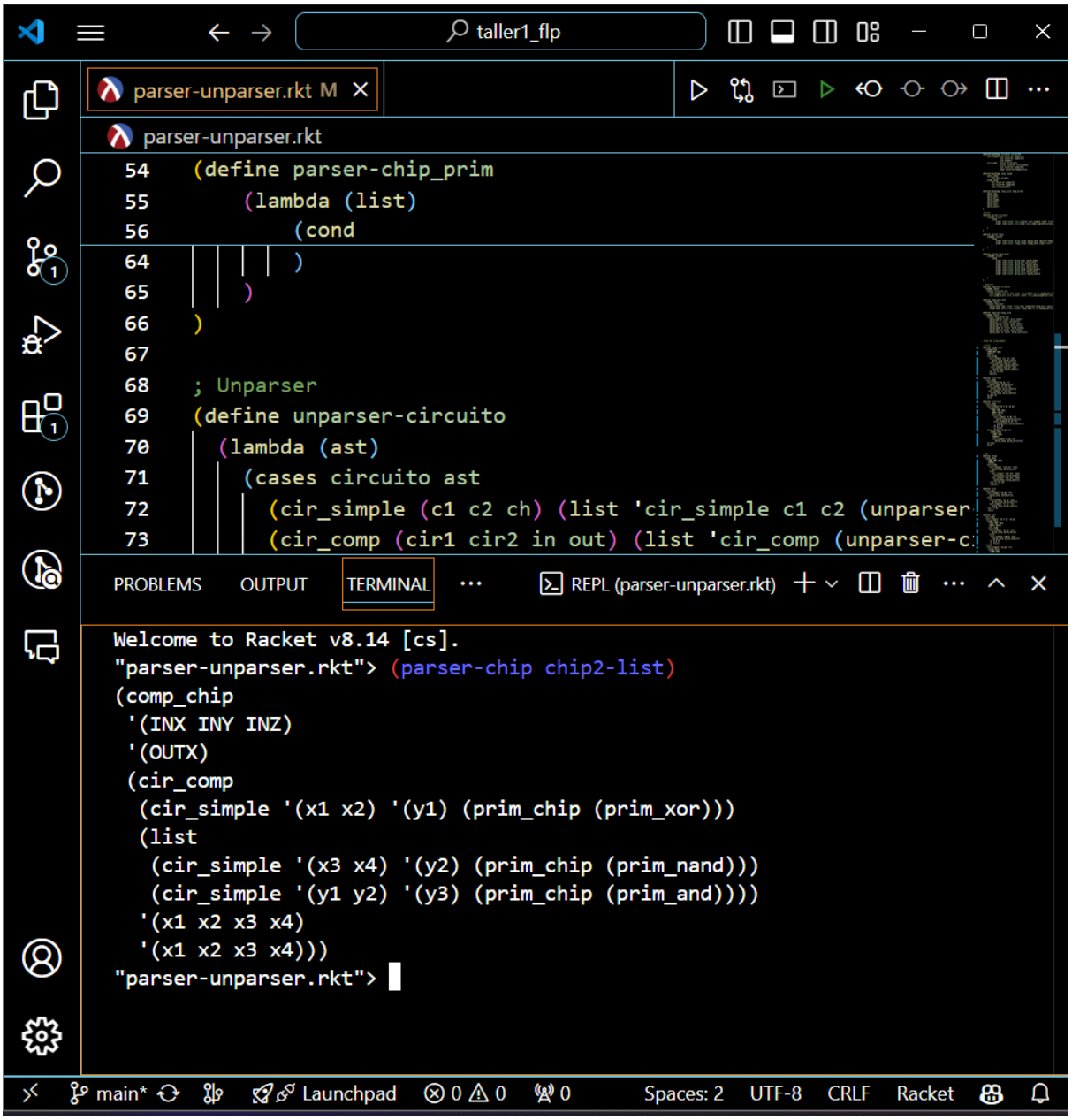
```
"taller1_flp\\representacion-procedimientos.rkt"> (chip2 0)
'comp_chip
"taller1_flp\\representacion-procedimientos.rkt"> (chip2 1)
'(INA INB)
"taller1_flp\\representacion-procedimientos.rkt"> (chip2 2)
'(OUTA OUTB)
"taller1_flp\\representacion-procedimientos.rkt"> (chip2 3)
#<procedure:...-procedimientos.rkt:16:8>
```

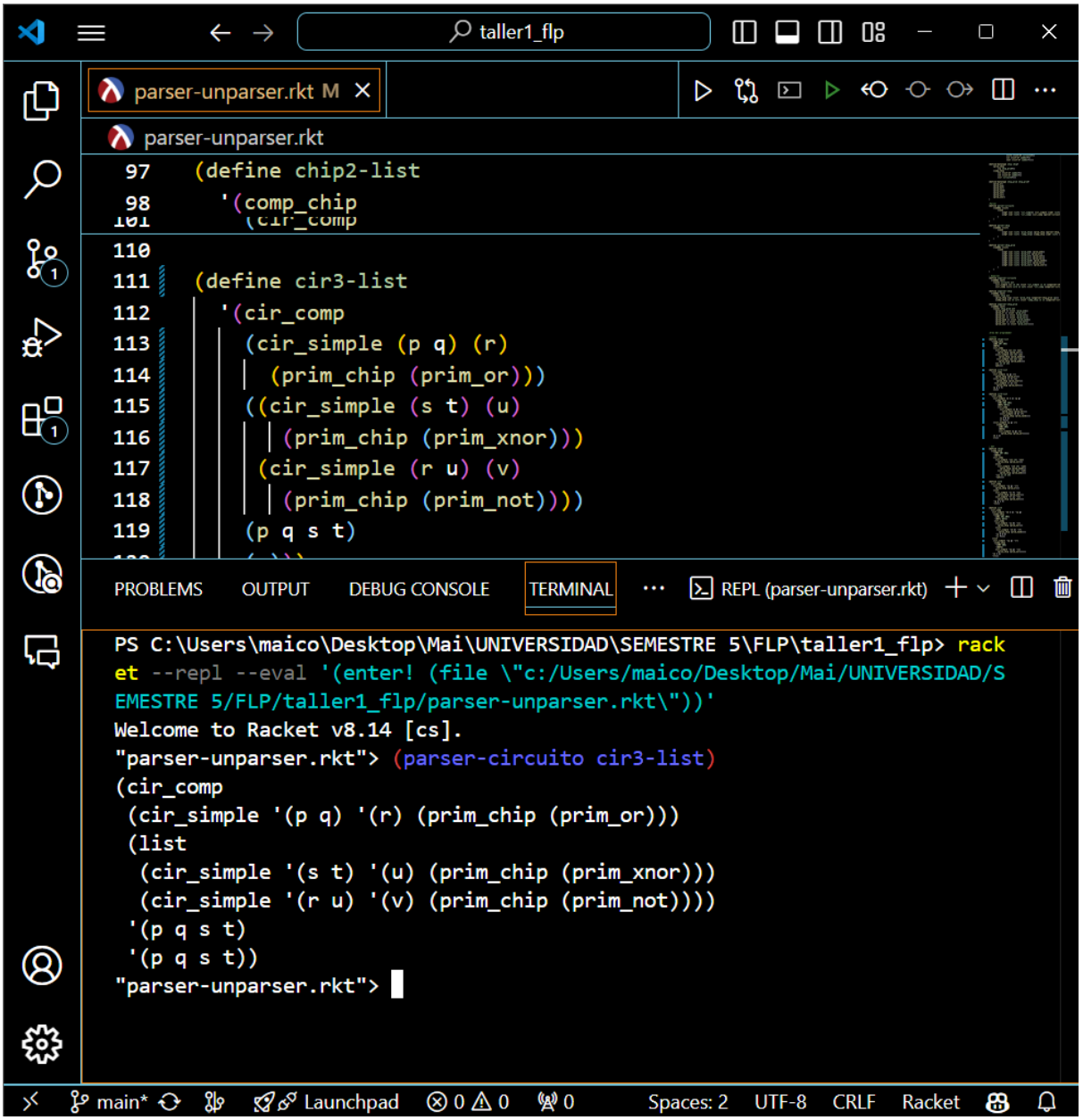
CAPTURAS DATATYPE

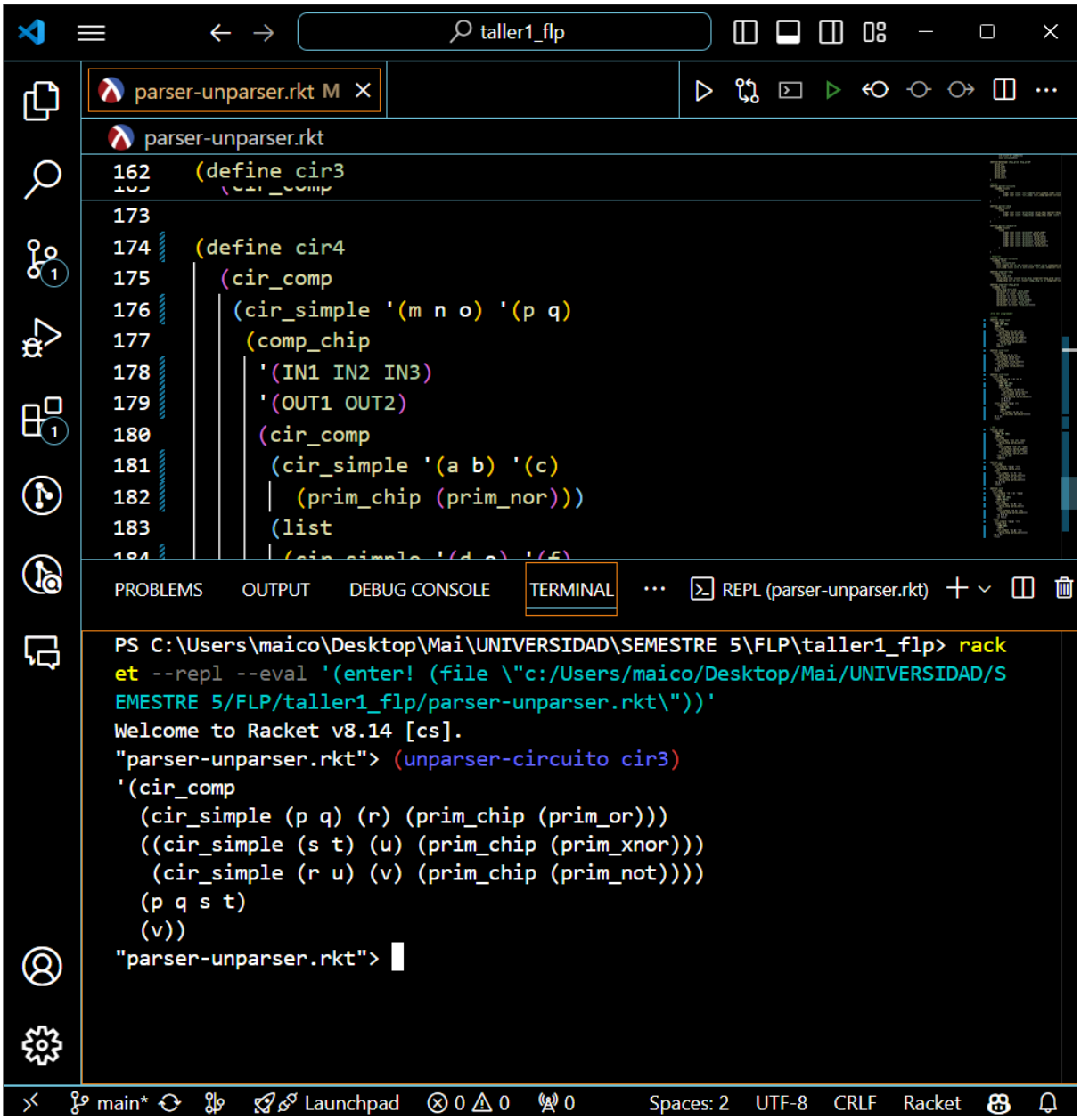


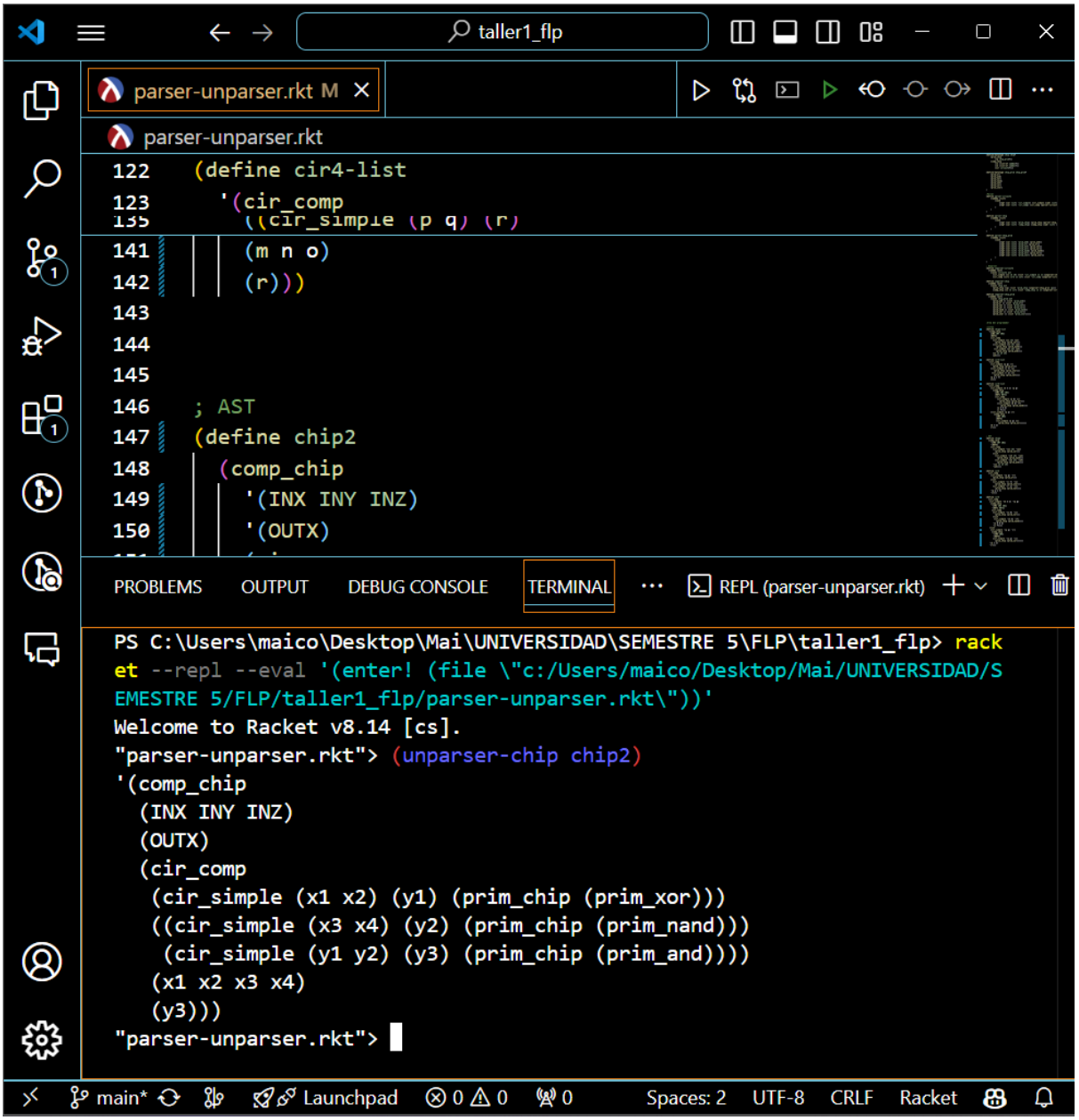


CAPTURAS PARSER-UNPARSER:


taller1_flp
parser-unparser.rkt M
parser-unparser.rkt
54 (define parser-chip_prim
55 (lambda (list)
56 (cond
64)
65)
66)
67
68 ; Unparser
69 (define unparser-circuito
70 (lambda (ast)
71 (cases circuito ast
72 (cir_simple (c1 c2 ch) (list 'cir_simple c1 c2 (unparser
73 (cir_comp (cir1 cir2 in out) (list 'cir_comp (unparser-c
PROBLEMS
OUTPUT
TERMINAL
REPL (parser-unparser.rkt)
Welcome to Racket v8.14 [cs].
"parser-unparser.rkt"> (parser-chip chip2-list)
(comp_chip
'(INX INY INZ)
'(OUTX)
(cir_comp
(cir_simple '(x1 x2) '(y1) (prim_chip (prim_xor)))
(list
(cir_simple '(x3 x4) '(y2) (prim_chip (prim_nand)))
(cir_simple '(y1 y2) '(y3) (prim_chip (prim_and))))
'(x1 x2 x3 x4)
'(x1 x2 x3 x4)))
"parser-unparser.rkt">
main*
Launchpad
0 0
0
Spaces: 2
UTF-8
CRLF
Racket


taller1_flp
parser-unparser.rkt M
parser-unparser.rkt
97 (define chip2-list
98 '(comp_chip
110
111 (define cir3-list
112 '(cir_comp
113 (cir_simple (p q) (r)
114 (prim_chip (prim_or)))
115 ((cir_simple (s t) (u)
116 (prim_chip (prim_xnor)))
117 (cir_simple (r u) (v)
118 (prim_chip (prim_not))))
119 (p q s t)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
REPL (parser-unparser.rkt)
PS C:\Users\maico\Desktop\Mai\UNIVERSIDAD\SEMESTRE 5\FLP\taller1_flp> rack
et --repl --eval '(enter! (file \"c:/Users/maico/Desktop/Mai/UNIVERSIDAD/S
EMESTRE 5/FLP/taller1_flp/parser-unparser.rkt\"))'
Welcome to Racket v8.14 [cs].
"parser-unparser.rkt"> (parser-circuito cir3-list)
(cir_comp
(cir_simple '(p q) '(r) (prim_chip (prim_or)))
(list
(cir_simple '(s t) '(u) (prim_chip (prim_xnor)))
(cir_simple '(r u) '(v) (prim_chip (prim_not))))
'(p q s t)
'(p q s t))
"parser-unparser.rkt">
main*
Launchpad
0 0
0
Spaces: 2
UTF-8
CRLF
Racket


taller1_flp
parser-unparser.rkt M
parser-unparser.rkt
162 (define cir3
173
174 (define cir4
175 (cir_comp
176 (cir_simple '(m n o) '(p q)
177 (comp_chip
178 '(IN1 IN2 IN3)
179 '(OUT1 OUT2)
180 (cir_comp
181 (cir_simple '(a b) '(c)
182 (prim_chip (prim_nor)))
183 (list
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
REPL (parser-unparser.rkt)
PS C:\Users\maico\Desktop\Mai\UNIVERSIDAD\SEMESTRE 5\FLP\taller1_flp> racket --repl --eval '(enter! (file \"c:/Users/maico/Desktop/Mai/UNIVERSIDAD/SEMESTRE 5/FLP/taller1_flp/parser-unparser.rkt\"))'
Welcome to Racket v8.14 [cs].
"parser-unparser.rkt"> (unparser-circuito cir3)
'(cir_comp
(cir_simple (p q) (r) (prim_chip (prim_or)))
((cir_simple (s t) (u) (prim_chip (prim_xnor)))
(cir_simple (r u) (v) (prim_chip (prim_not))))
(p q s t)
(v))
"parser-unparser.rkt">
main*
Launchpad
Spaces: 2
UTF-8
CRLF
Racket


taller1_flp
parser-unparser.rkt M
parser-unparser.rkt
122 (define cir4-list
123 '(cir_comp
141 (m n o)
142 (r)))
143
144
145
146 ; AST
147 (define chip2
148 (comp_chip
149 '(INX INY INZ)
150 '(OUTX)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
REPL (parser-unparser.rkt)
PS C:\Users\maico\Desktop\Mai\UNIVERSIDAD\SEMESTRE 5\FLP\taller1_flp> racket --repl --eval '(enter! (file \"c:/Users/maico/Desktop/Mai/UNIVERSIDAD/SEMESTRE 5/FLP/taller1_flp/parser-unparser.rkt\"))'
Welcome to Racket v8.14 [cs].
"parser-unparser.rkt"> (unparser-chip chip2)
'(comp_chip
(INX INY INZ)
(OUTX)
(cir_comp
(cir_simple (x1 x2) (y1) (prim_chip (prim_xor)))
((cir_simple (x3 x4) (y2) (prim_chip (prim_nand)))
(cir_simple (y1 y2) (y3) (prim_chip (prim_and))))
(x1 x2 x3 x4)
(y3)))
"parser-unparser.rkt">
main*
Launchpad
Spaces: 2
UTF-8
CRLF
Racket