

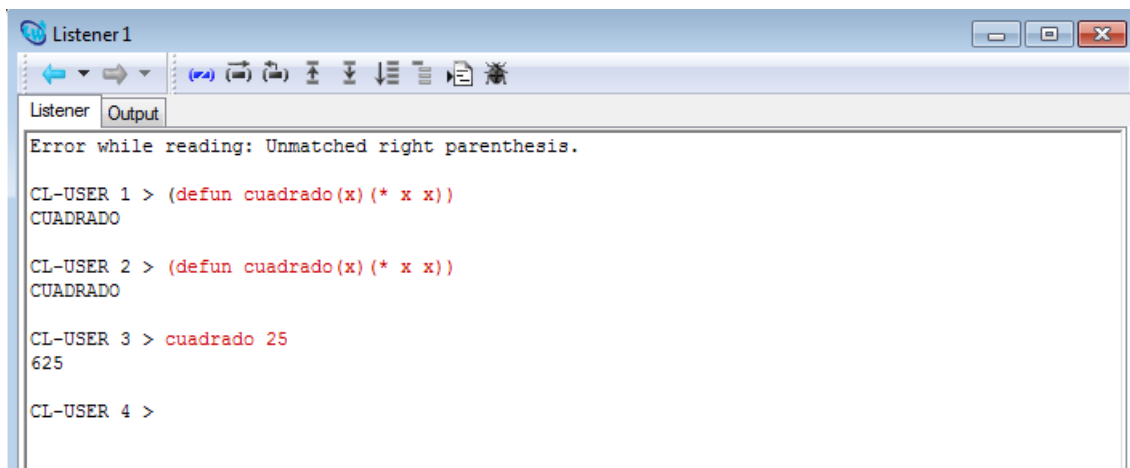
EJERCICIOS COMANDO DEFUN

EJERCICIO 1:

El comando **defun** define una función personalizada:

```
CL-USER 2 > (defun cuadrado(x) (* x x))  
CUADRADO  
  
CL-USER 3 > cuadrado 25  
625
```

Resultado en LISPWORK:



❖ Función valor absoluto:

```
(defun absoluto (n)(cond ((> n 0) n)((- n))))
```

```
(absoluto -45)
```

Resultado en LISPWORK:

```
CL-USER 1 > (defun f-primer0 (x y) x)
F-PRIMERO

CL-USER 2 > f-primer0 3 (print 4)

4
3

CL-USER 3 > (defun absoluto (n) (cond ((> n 0) n) ((- n))))
ABSOLUTO

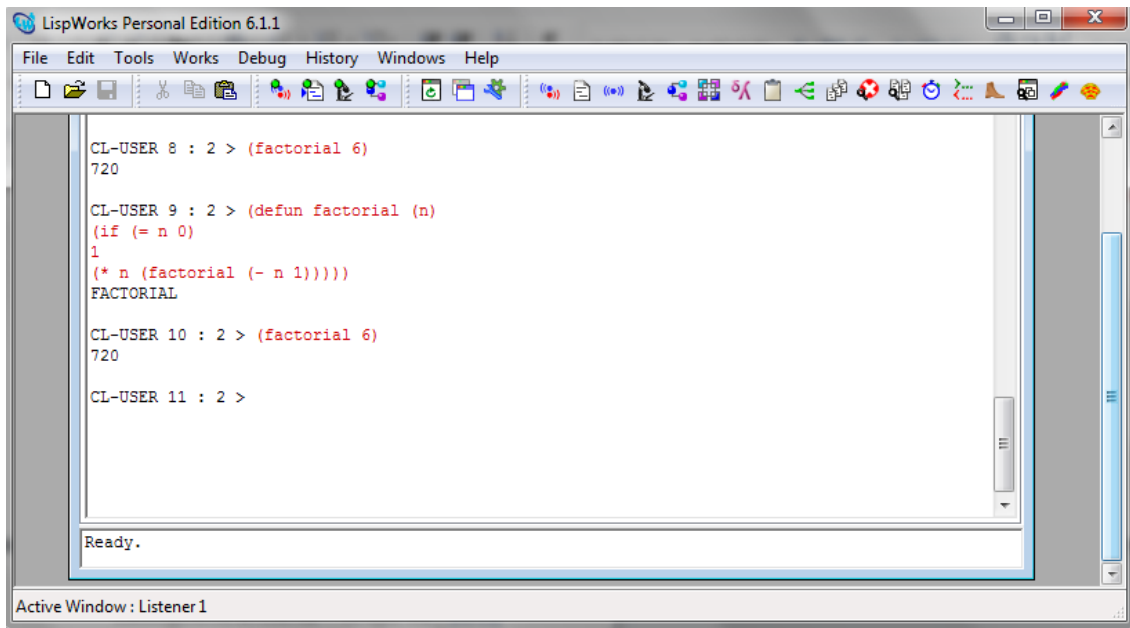
CL-USER 4 > (absoluto -45)
45

CL-USER 5 >
```

❖ Función Factorial:

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1)))))
```

Resultado en LISPWORK:



EJERCICIO 2:

Se define **if** con una condición y dos posibles opciones:

```
(defun mayor(x)(if (> (car x) (cadr x))(list (car x))(list(cadr x)) ))
```

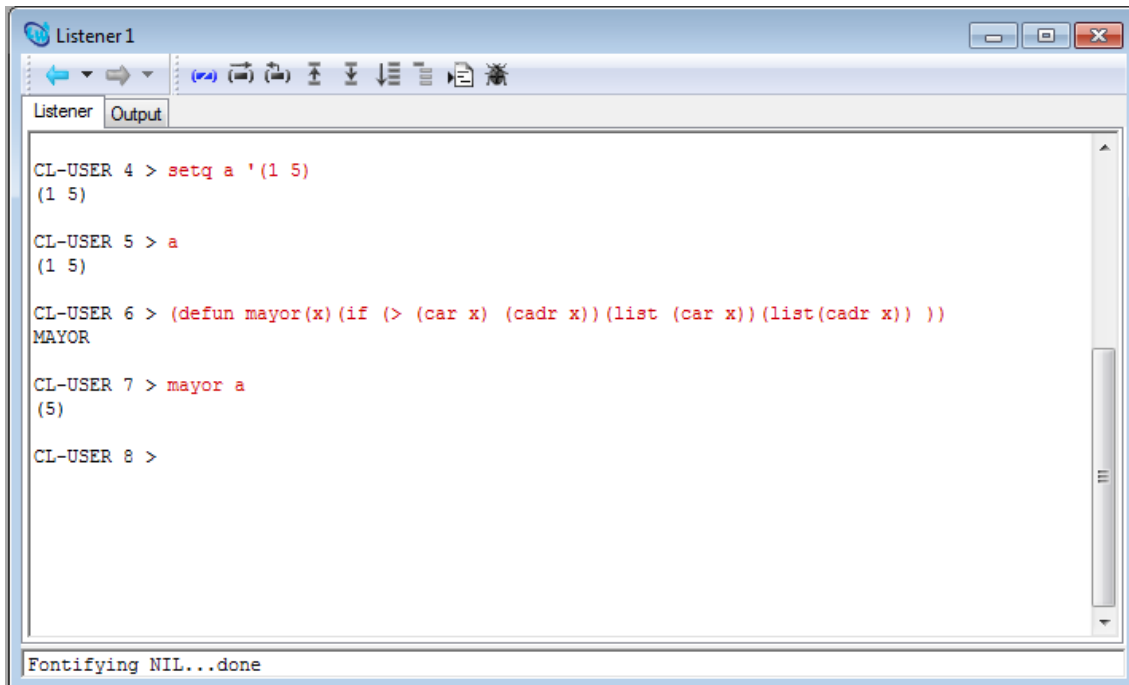
```
CL-USER 4 > setq a '(1 5)
(1 5)
```

```
CL-USER 5 > a
(1 5)
```

```
CL-USER 6 > (defun mayor(x)(if (> (car x) (cadr x))(list (car x))(list(cadr x)) ))
MAYOR
```

```
CL-USER 7 > mayor a
(5)
```

Resultado en LISPWORK:



```

Listener 1
Listener Output

CL-USER 4 > setq a '(1 5)
(1 5)

CL-USER 5 > a
(1 5)

CL-USER 6 > (defun mayor(x) (if (> (car x) (cadr x)) (list (car x)) (list (cadr x))))
MAYOR

CL-USER 7 > mayor a
(5)

CL-USER 8 >

Fontifying NIL...done
  
```

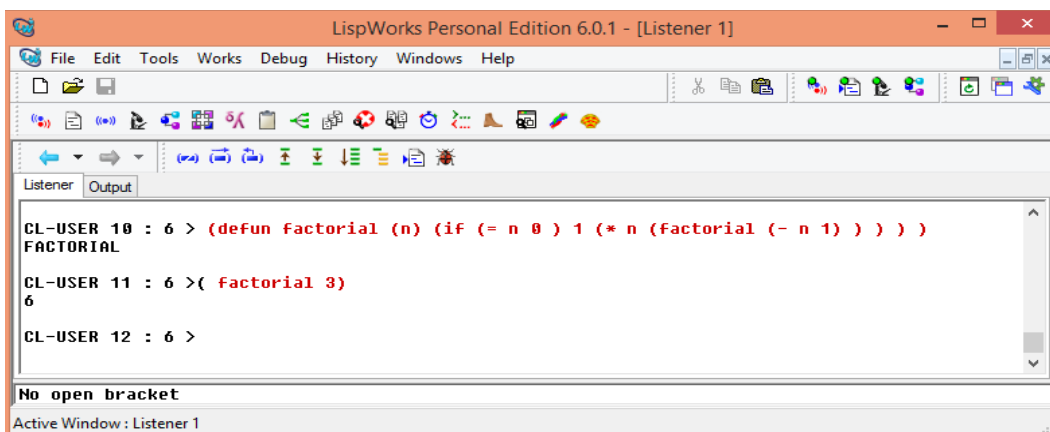
EJERCICIO 3:

Construya una función recursiva para hallar el factorial de n.

Solución:

```
(defun factorial (n) (if (= n 0) 1 (* n (factorial (- n 1)))))
```

En lisp:



```

LispWorks Personal Edition 6.0.1 - [Listener 1]
File Edit Tools Works Debug History Windows Help

CL-USER 10 : 6 > (defun factorial (n) (if (= n 0) 1 (* n (factorial (- n 1)))))
FACTORIAL

CL-USER 11 : 6 > (factorial 3)
6

CL-USER 12 : 6 >

No open bracket
Active Window : Listener 1
  
```

EJERCICIO 4:

Construya una función recursiva si encuentra un numero dentro de una lista, lo retorne, caso contrario, indique nil.

Solución

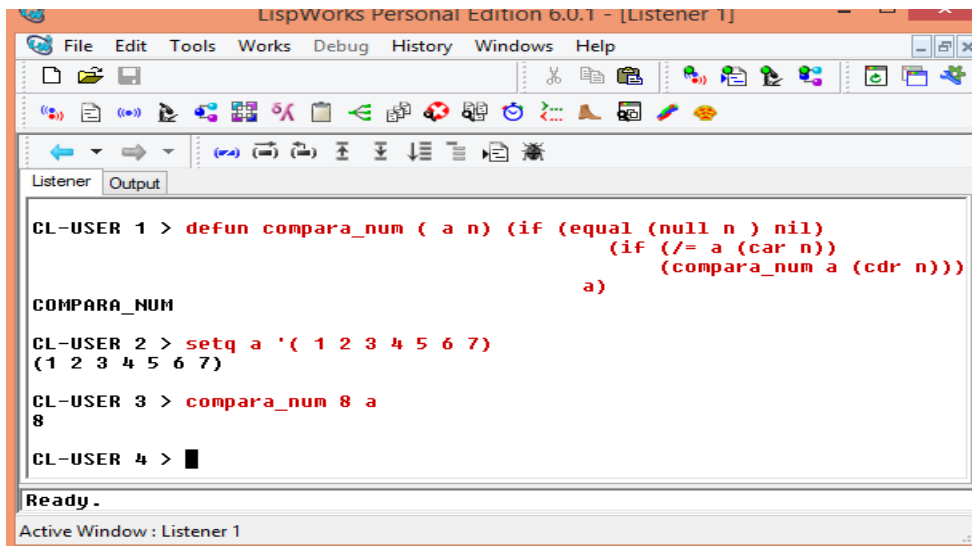
```
defun compara_num ( a n) (if (equal (null n ) nil)

                        (if (/= a (car n))

                            (compara_num a (cdr n)))

                        a)
```

En lisp:



The screenshot shows the LispWorks Personal Edition 6.0.1 interface. The main window displays the following code and output:

```
CL-USER 1 > defun compara_num ( a n) (if (equal (null n ) nil)
                                          (if (/= a (car n))
                                              (compara_num a (cdr n)))
                                          a)

COMPARA_NUM

CL-USER 2 > setq a '( 1 2 3 4 5 6 7)
(1 2 3 4 5 6 7)

CL-USER 3 > compara_num 8 a
8

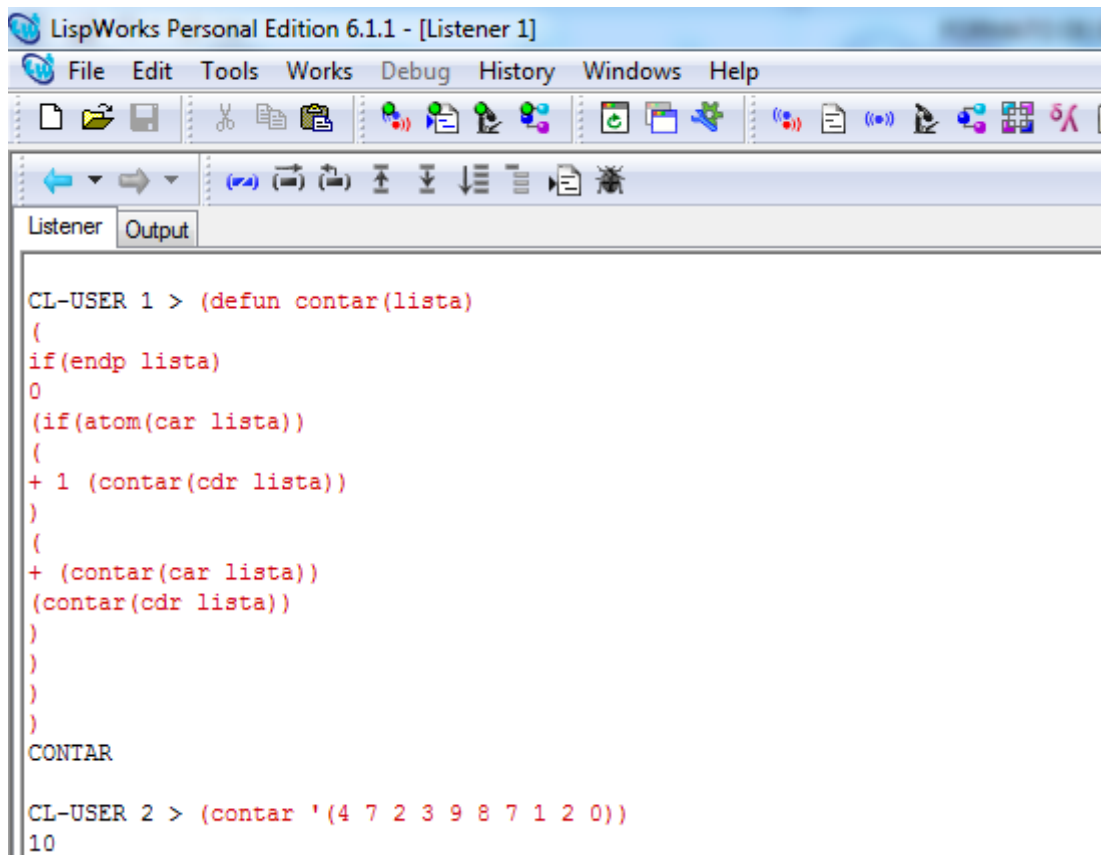
CL-USER 4 > █

Ready .
Active Window : Listener 1
```

EJERCICIO 5:

Hacer una función que cuente todos los elementos de una lista.

Ejemplo: CONTAR (1 2 3 3 2 9) = 6.



LispWorks Personal Edition 6.1.1 - [Listener 1]

File Edit Tools Works Debug History Windows Help

Listener Output

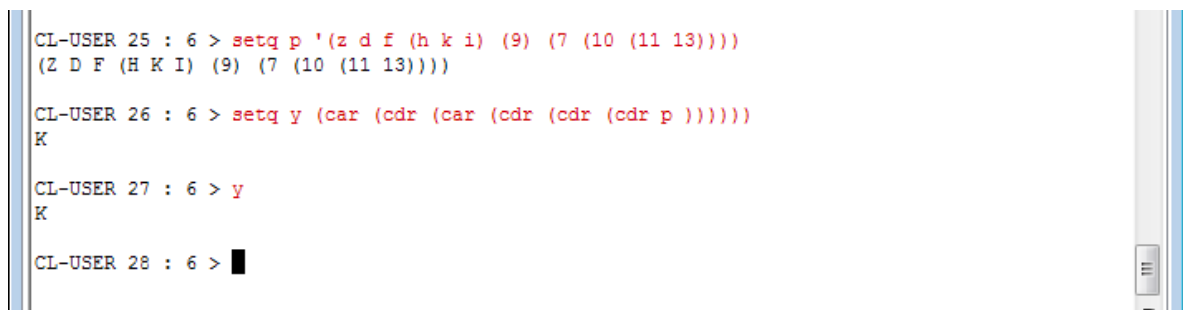
```
CL-USER 1 > (defun contar(lista)
  (
    if(endp lista)
    0
    (if(atom(car lista))
      (
        + 1 (contar(cdr lista))
      )
      (
        + (contar(car lista))
          (contar(cdr lista))
      )
    )
  )
)
CONTAR

CL-USER 2 > (contar '(4 7 2 3 9 8 7 1 2 0))
10
```

EJERCICIO 6:

Hacer una función que devuelva el último átomo de una lista.

Ejemplo: ULTIMO (3 2 3 7 4 6 8 1 9) = 9.

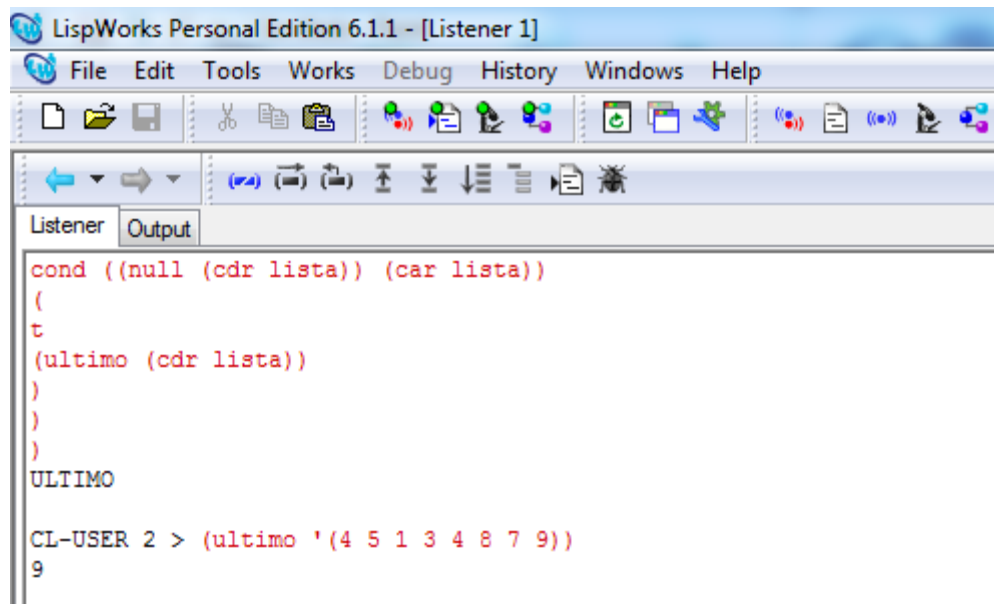


```
CL-USER 25 : 6 > setq p '(z d f (h k i) (9) (7 (10 (11 13))))
(Z D F (H K I) (9) (7 (10 (11 13))))

CL-USER 26 : 6 > setq y (car (cdr (car (cdr (cdr (cdr p))))))
K

CL-USER 27 : 6 > y
K

CL-USER 28 : 6 > █
```



```

LispWorks Personal Edition 6.1.1 - [Listener 1]
File Edit Tools Works Debug History Windows Help

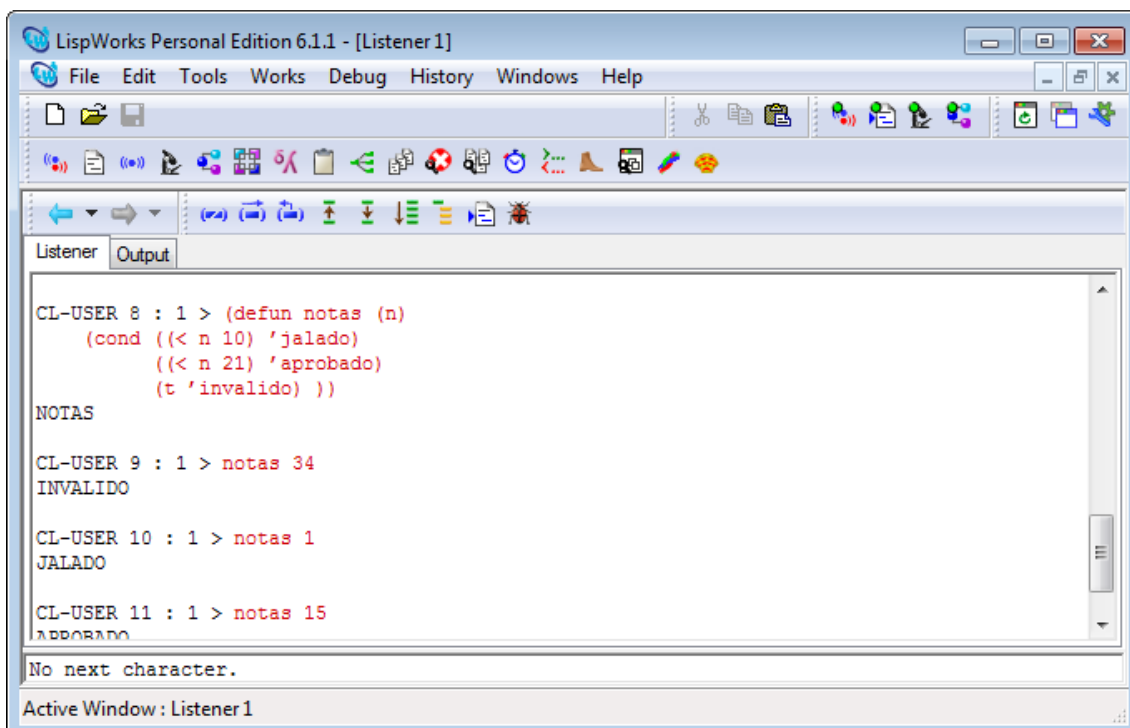
cond ((null (cdr lista)) (car lista))
(
  t
  (ultimo (cdr lista))
)
)
)
ULTIMO

CL-USER 2 > (ultimo '(4 5 1 3 4 8 7 9))
9

```

EJERCICIO 7:

Se quiere evaluar si una nota es aprobada o jalada, si la nota es menos a 11 el alumno esta jalado, sino está aprobado, cualquier otro número ingresado de forma errónea regresara el enunciado “invalido”



```

LispWorks Personal Edition 6.1.1 - [Listener 1]
File Edit Tools Works Debug History Windows Help

CL-USER 8 : 1 > (defun notas (n)
  (cond ((< n 10) 'jalado)
        ((< n 21) 'aprobado)
        (t 'invalido) ))
NOTAS

CL-USER 9 : 1 > notas 34
INVALIDO

CL-USER 10 : 1 > notas 1
JALADO

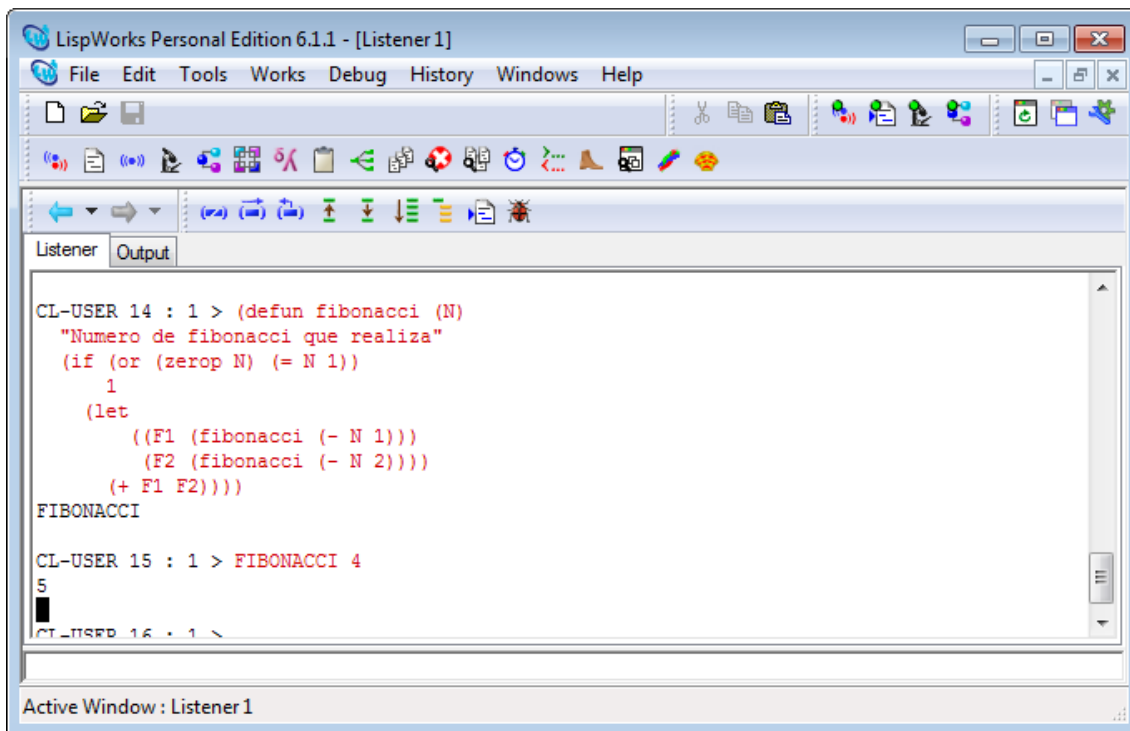
CL-USER 11 : 1 > notas 15
APROBADO

No next character.
Active Window: Listener1

```

EJERCICIO 8

Fibonacci



The screenshot shows the LispWorks Personal Edition 6.1.1 - [Listener 1] window. The interface includes a menu bar (File, Edit, Tools, Works, Debug, History, Windows, Help), a toolbar with various icons, and a main text area. The text area contains the following code:

```
CL-USER 14 : 1 > (defun fibonacci (N)
  "Numero de fibonacci que realiza"
  (if (or (zerop N) (= N 1))
      1
      (let
        ((F1 (fibonacci (- N 1)))
         (F2 (fibonacci (- N 2))))
        (+ F1 F2))))
FIBONACCI

CL-USER 15 : 1 > FIBONACCI 4
5
CL-USER 16 : 1 >
```

The status bar at the bottom indicates "Active Window : Listener 1".

EJERCICIO 9

EJERCICIO 10