

TEMA 2: EJECICIOS COMMON LISP

HUMBERTO BUSTINCE
Dpto. Automática y Computación
bustince@unavarra.es

- **El programador en LISP encierra sus pensamientos en nidos de paréntesis (Alan Perlis)**

1. Dibuja la representación de los símbolos y las estructuras y punteros de los CONSES para el ejemplo, (ANIMAL (CASA-MASCOTAS (PERRO GATO)))

2. Representa las listas de verdad y listas punteadas siguientes:
(SEMANA FIN-DE-SEMANA SABADO.DOMINGO)
(SEMANA FIN-DE-SEMANA (SABADO.DOMINGO))
((LAS-LISTAS.PUEDEN) TENER VARIAS ESTRUCTURAS)

3. Representa las siguientes listas con CONSES.
()

(B)

(1 2 3 4 5)

(A (B C))

HURACAN-ALICIA
((V-A-KmH 170)(HUMEDAD 100)(LOCALIZACION (NORTE 32.8)))

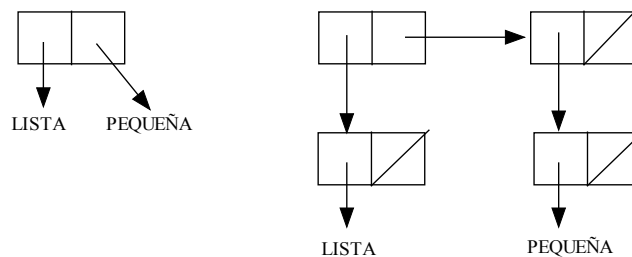
DIAS-LABORABLES
(LUN MAR MIE JUE VIE)

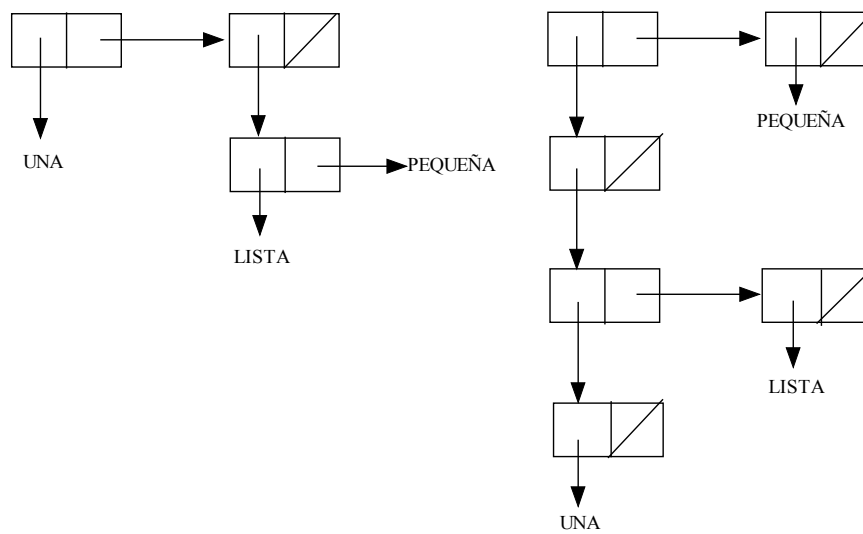
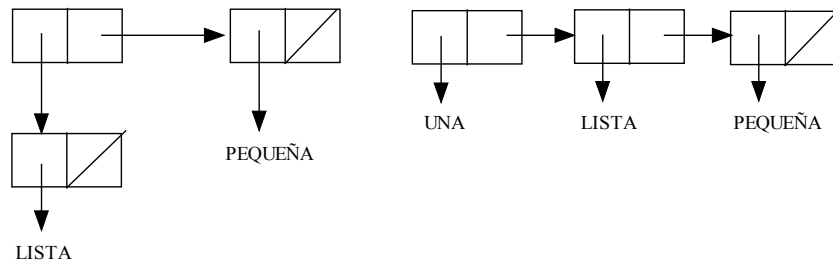
DIAS-FUERA
(LUN MIE)

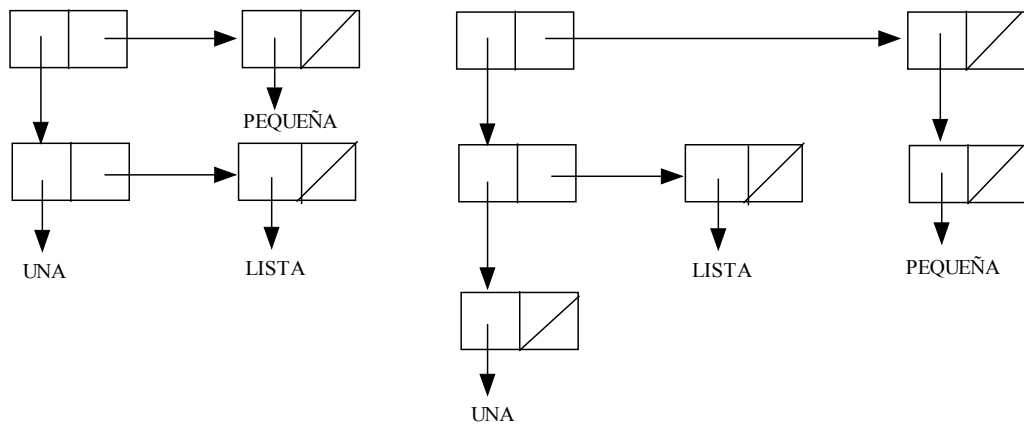
((LAS-LISTAS PUEDEN) TENER VARIAS ESTRUCTURAS)

(LU MA MI JU VI SEMANA SABADO.DOMINGO FIN-DE-SEMANA)

4. Escribe los siguientes gráficos como listas.







5. Representa mediante casillas CONS las siguientes listas Lisp.

- a) (pedro pepe pablo)
- b) (pepe.pablo)
- c) (marcos.(esteban.juan))
- d) (ruth(mirian.olga))
- e) ((marlene.elisabeth).irene)

f) ((arroz.pasta).(sopa.(caldo.carne)))

g) (((chocolate.tarta).mermelada).nata)

h) ((x) y z)

6. ¿Qué ocurre en Lisp con las siguientes instrucciones?. ¿Por qué?

➤ 3 <return>

➤ “esto es un string” <return>

➤ t <return>

➤ A <return>

➤ ‘A<return>

➤ ‘(+ 2 4) <return>

➤ (+ 2 4) <return>

➤ (PRINT ‘X) <return>

➤ (PRINT “HOLA”) <return>

➤ (PRINC “HOLA”) <return>

➤ (*(+ 2 3)(/ 12 4)) <return>

➤ (QUOTE X) <return>

- (QUOTE (A)) <return>
- '(A) <return>
- '(A B C) <return>
- '(PALABRAS (CONSTRUYE GATOS PERROS)) <return>
- (CAR (A B C)) <return>
- (CAR '(A B C)) <return>
- (CAR (A B C)) <return>
- (CAR '(A B (C D) E)) <return>
- (CAR '((A B) C)) <return>
- (CAR '((A B C))) <return>
- (CAR 'A) <return>
- (CDR '(A B C)) <return>
- (CDR '(A B (C D) E)) <return>
- (CDR '((A B) C)) <return>
- (CDR (1 2 3 4)) <return>
- (CDR '((A))) <return>
- (CDR '((A B C))) <return>
- (CDR 'A) <return>
- (CADR '(A B C)) <return>
- (CDAR '((A B C) D E)) <return>

- (CADDR '(A B C D E)) <return>
- (LAST '(A B C)) <return>
- (LAST '(A (B C))) <return>
- (LAST '(1 2 3.5)) <return>
- (NTH 1 '(A B C)) <return>
- (CONS "HOLA" ("COMO" "ESTAS")) <return>
- (CONS "HOLA" (LIST "COMO" "TE" "LLAMAS")) <return>
- (CONS 'A '(B C D)) <return>
- (CONS '(X Y) '(B D C)) <return>
- (LIST 'A 'B 'C) <return>
- (LIST '(A) '(B) '(C)) <return>
- (LIST(LIST(LIST 'A))) <return>
- (SETF L '(1 2)) <return>
- (SETF K '(3 4)) <return>
- (APPEND L K) <return>
- L<return>
- K<return>
- (APPEND '(A) '(B) '(C)) <return>
- (APPEND '(A) '(BC)) <return>
- (APPEND '(A) '(B) 'C) <return>

- (APPEND 'A '(B C)) <return>
- (APPEND 'A NIL) <return>
- (SETF A '(1 2 3)) <return>
- (RPLACA A 7) <return>
- (RPLAD A '(9)) <return>
- (SETF A '(1 2 3)) <return>
- (SETF B '(4 5 6)) <return>
- (NCONC A B) <return>
- A <return>
- B <return>
- (SETF B '(4 5 6 7)) <return>
- (PUSH 9 B) <return>
- B <return>
- (POP B) <return>
- B <return>
- (LENGTH '(A B C)) <return>
- (LENGTH '(A B.C)) <return>
- (LENGTH '()) <return>
- (MEMBER 'C '(A B C D E F)) <return>

➤ (MEMBER 'Z '(A B C D E F)) <return>

➤ (APPEND '(1 2) '((3 4)) (LIST 5 6)) <return>

7.

➤ (SETF X '(1 (B C) A)) <return>

➤ (SETF Y '(1 2 3.4)) <return>

➤ (SETF Z '(3 7))

➤ (APPEND X Y Z) <return>

8.

➤ (CADDAR '("HOLA" "QUE" "TAL") "COMO" "TE" "LLAMAS")) <return>

➤ (NTH 3 '("HOLA" "COMO" "TE" "LLAMAS")) <return>

➤ (MEMEBER 'X '(A X B X C)) <return>

➤ (MEMBER 'X '(A (X) B)) <return>

➤ (SETF L '((A B) (C D))) <return>

➤ (MEMBER '(C D) L) <return>

➤ (MEMBER 2.0 '(1 2 3)) <return>

➤ (MEMBER '(C D) L:TEXT #'EQUAL) <return>

➤ (MEMBER 2.0 '(1 2 3) :TEST #'=) <return>

➤ (MEMBER 2.0 '(1 2 3) :TEST #'<) <return>

9.

- (+ 2 3 4) <return>
- (- 10 7 3) <return>
- (* 3 4 0.5) <return>
- (/ 100 4 5) <return>
- (/ 12 9) <return>
- (/ 18 7) <return>
- (/ 25.0 10) <return>
- (/ 7) <return>
- (/48 24 (* 2 3)) <return>
- (SETF C 5) <return>
- (INCF C 10) <return>
- C <return>
- (SETF C 9) <return>
- (DECF C 3) <return>
- C <return>
- (SETF C 5) <return>
- (INCF C) <return>
- C <return>
- (SETF E 7) <return>
- (DECF E) <return>
- E <return>

- (SETF C 3) <return>
- (1+ C) <return>
- C<return>
- (SETF L 2) <return>
- (1- L) <return>
- L <return>
- (SQRT 16) <return>
- (SQRT 2) <return>
- (EXPT 2 3) <return>
- (EXP 1) <return>
- (REM 5 3) <return>
- (MOD 5 3) <return>
- (ABS -5) <return>
- (MIN 2 4 7 1 5) <return>
- (MAX 2 4 7 1 5) <return>
- (SETF N 100) <return>
- (RANDOM N) <return>
- (= 3 3.0) <return>
- (/ 3 3.0) <return>
- (< 3 5 8) <return>
- (< 3 5 4) <return>
- (> 8 5 3) <return>
- (> 8 3 5) <return>
- (<= 5 5 8) <return>
- (<= 9 9 4) <return>

- (\geq 9 7 7) <return>
- (\geq 8 9 8) <return>
- (NUMBERP 7) <return>
- (NUMBERP 'NOMBRE) <return>
- (ODDP -7) <return>
- (ODDP 5.8) <return>
- (EVENP 8) <return>
- (EVENP 'NOMBRE) <return>
- (= 7.5 7.5) <return>
- (EQ 7.5 7.5) <return>
- (EQ 3 3) <return>
- (EQ 2 2.0) <return>
- (EQL 7.3 7.3) <return>
- (EQL 7 7.0) <return>
- (EQUAL 7.5 7.5) <return>
- (EQUAL '(1 2) '(1 2)) <return>
- (EQUAL "HOLA" "HOLA") <return>
- (EQUAL "HOLA" "Hola") <return>
- (EQUAL 7 7.0) <return>
- (EQUALP "HOLA" "Hola") <return>
- (EQUALP 7 7.0) <return>
- (= "HOLA" "HOLA") <return>
- (= 2.5 5/2 (- 4 1 0.5)) <return>
- (ATOM 'UNA) <return>

- (ATOM ())<return>
- (ATOM "STRING") <return>
- (LISTP NIL) <return>
- (SETF L 'WORD) <return>
- (SETF K 'WORD) <return>
- (EQUAL L K) <return>
- (SETF L '(A B C)) <return>
- (EQUAL '(A B C) L) <return>
- (EQL '(A B C) L) <return>

10.

- (SETF L '(A B C)) <return>
- (SETF M L) <return>
- (SETF N '(A B C)) <return>

Contesta a las siguientes preguntas:

- (EQUAL L M) <return>
- (EQUAL L N) <return>
- (EQUAL '(A B C) L) <return>
- (EQL L M) <return>
- (EQL L N) <return>
- (EQL '(A B C) L) <return>

11.

- (AND T T T (* 2 5)) <return>
 - (SETF X 3) <return>
 - (SETF CONT 0) <return>
 - (INCF CONT) <return>
 - (AND (<= CONT 10)(NUMBERP X)(* 2 X)) <return>
 - (AND (EVENP X) (/ X 2)) <return>
-
- (OR NIL NIL (NULL '(A B) (REM 23 13))) <return>
 - (SETF X 10) <return>
 - (OR (< 0 X)(DECF X)) <return>
-
- (NOT NIL) <return>
 - (NOT T) <return>
-
- (NOT (EQUAL 'A 'A)) <return>
 - (SETF X '(A B C)) <return>
-
- (AND (NOT (NUMBERP X))(NOT (ATOM X))"ESTO ES UN EJEMPLO") <return>
-
- (OR (< 2 3) (SETF PRUEBA 1) NIL) <return>
 - PRUEBA <return>
 - (AND(SETF PRUEBA 1) NIL) <return>
 - PRUEBA <return>

12.

➤ (SETF X 5) <return>

➤ (SETF Y 4) <return>

➤ (IF (> Y X) Y X) <return>

> (COND ((= Y X) "IGUALES")
 ((< X Y) "MENOR")
 (t "MAYOR")) <return>

13. Construir un programa que a partir de la temperatura de un recinto devuelva,

- a) HELADO si la temperatura es menor de 0 grados.
- b) FRIO si está entre 10 y 20.
- c) CALIDO si está entre 20 y 30.
- d) ABRASADOR si es mayor de 30.

14. (SETF X 5)<return>

(when (< x 10) (incf x) (princ x) (princ "hola") (* 10 10))<return>

(when (< x 10) (incf x) (print x) (print "hola") (* 10 10))<return>

15. Leer un mes y un año y utilizando la estructura CASE devolver los días de ese mes.

16.

- (SETF X '(1 2 3 4 5)) <return>
- (SETF L '())<return>
- (SETF CONTADOR 0) <return>

```
(LOOP
  (IF (= (1+ CONTADOR)(LENGTH X))(RETURN L)
    (PROGN (NCONC L (1+ (NTH CONTADOR X)))
            (INCF CONTADOR))))
```

17.

```
(DO ((X 1 (+ X 1))(Y 1 (* Y 2)))(> X 5) Y)
  (PRINT X))
```

18. Escribe utilizando LOOP la siguiente secuencia de números

1 2 3 4 5
5

19.

➤ (MAPCAR #' + '(1 3 4) '(5 6 7)) <return>

➤ (MAPCAR #' ODDP '(1 5 6)) <return>

➤ (MAPCAR #' * '(2 4) '(6 7) '(2 5)) <return>

17. Escribe utilizando DO los siguiente:

“HOLA”
“HOLA”
“HOLA”
“HOLA”
“HOLA”
“FINAL”

20. Utilizando la iteración DO escribe un programa que haga los siguientes dibujos.

a) 1
2 2
3 3 3
4 4 4 4
FINAL

b) 1
1 2
1 2 3
1 2 3 4
FINAL

c) 1
1 1
1 1 1
1 1 1 1
FINAL

d) 1
1 1
1 1 1
1 1 1 1
FINAL

21.

```
>(DOLIST (X '(1 2 3 4 5))  
  (PRINC (+ X 10))  
  (PRINC #\ )) <return >
```

```
>(DOLIST ( X '(1 2 3 4 5))  
  (+ X 10)  
  (PRINC #\ ))
```

22. Utilizando DOLIST hacer un programa que dibuje:

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
NIL
```

23.

➤ (SETF X 'P) <return>

> (DOLIST (Y '(A B C P Q))
 (IF (EQUALP X Y) (RETURN 'ESTA))) <return>