REGLAS DE ESTILO LISP

CODIFICA UTILIZANDO ESTAS REGLAS.

Funciones

- Escribe funciones cortas que realicen una única operación bien definida.
- Utiliza nombres de funciones y variables que sean descriptivos. Es decir, que ilustren para qué son utilizadas y sirvan así de documentación.

Formato

- Utiliza adecuadamente la sangría.
- No escribas líneas demasiado largas
- Utiliza los espacios en blanco para separar segmentos de código, pero no abuses de ellos

ERRÓNEO 1:

```
(defun foo(x y)(let((z(+ x y 10)))(* z z)))
```

ERRÓNEO 2:

```
(defun foo ( x  y )
  (let ( ( z (+ x y 10) ) )
    ( * z z )
    )
  )
```

CORRECTO:

```
(defun foo (x y)
  (let ((z (+ x y 10)))
    (* z z)))
```

Recomendaciones para codificar

- Utiliza preferentemente recursión.
- Aprende a usar funciones propias de LISP: ƒ MAPCAR y MAPCAN (operaciones en paralelo)
- Trata de no usar operaciones destructivas: NCONC, NREVERSE, DELETE. Usa las alternativas no destructivas: APPEND, REVERSE, REMOVE
- Cuando se trata de una lista, recuerda las alternativas FIRST/REST/SECOND/THIRD … en lugar de CAR/CDR/CADR/CADDR …

Condicionales

- Utiliza IF cuando la decisión tenga 2 ramas.
- Utiliza COND cuando la decisión tenga una o más ramas.
- Utiliza COND en lugar de PROGN dentro de una condición IF
- Utiliza COND en lugar de un conjunto de IFs anidados.

En caso de que haya varias alternativas, utiliza la más específica.

- No utilices EQUAL si EQL o EQ es suficiente.
- No utilices LET* si es suficiente con LET.
- Si una función es un predicado, debe devolver T/NIL. ƒ Si una función no es un predicado, debe devolver un valor útil.

Comenta el código

- Usa ;;; para explicaciones generales sobre bloques de código.
- Usa ;; para explicaciones en línea aparte antes de una línea de código.
- Usa ; para explicaciones en la misma línea de código.

REGLAS DE ESTILO LISP

ERRORES DE CODIFICACIÓN A EVITAR

- Contempla siempre el caso NIL como posible argumento.
- Las funciones que se codifiquen pueden fallar, pero en ningún caso deben entrar en recursiones infinitas.
- Cuando las funciones de LISP fallan, el intérprete proporciona un diagnóstico que debe ser leído e interpretado. Ejemplo:

```
>  (rest 'a)
    Error: Attempt to take the cdr of A which is not listp.
    [condition type: TYPE-ERROR]
```

-
- Hacer una descomposición funcional adecuada. Puede que se necesiten funciones adicionales a las del enunciado.
- Diseña casos de prueba completos, no sólo los del enunciado.
- Si ya existen, utiliza las funciones de LISP, en lugar de codificar las propias (a menos que se indique lo contrario).
- Programad utilizando el sentido común (no intentando adivinar qué quiere el profesor). Pregunta si hay algo en el diseño de una función que no entiendes.