

11.2. AUTOLISP DE AUTOCAD.

El lenguaje AutoLISP se ofrece en AutoCAD como entorno de desarrollo a nivel interpretado; se trata de una adaptación de la variante **Common Lisp**, de la que se ha seleccionado un juego reducido de funciones de programación según las pautas generales del apartado 11.1.1 y una única estructura de datos, la lista¹. Existen operaciones para manipular listas según las referencias genéricas del apartado 11.1.2. Lo que le da un carácter especial al lenguaje es que no sólo los datos estructurados sino también los procedimientos que se programan se articulan en forma de lista.

Un conjunto de datos separados por espacios en blanco y delimitados por paréntesis constituyen una lista de datos; así, (1 4.5 "abc" (1 -34) 23) y (12.3 34.5 0.0) constituyen ejemplo de listas de datos; la primera es heterogénea y la segunda responde a un dato de tipo punto, pero ambas se construyen con las mismas funciones de programación. **En AutoLISP no existe declaración de tipos de variable**; éstas se adaptan automáticamente al tipo de la expresión que les da valor.

Una función de biblioteca tiene la expresión genérica (**NOMBRE arg1 arg2 ...**), donde NOMBRE es una palabra reservada y arg1, arg2, etc. dependen del cometido que tenga la función; obsérvese que la función adopta la estructura de una lista, que se distingue de la de datos por dos razones:

- **Su primer elemento es una palabra reservada** (un nombre de función).
- **Una función LISP realiza siempre un doble trabajo**: efectúa su cometido (si es una asignación, asigna; si es un ciclo, efectúa las vueltas; si es un IF, efectúa la condición, etc.) pero además, **indefectiblemente devuelve un valor al exterior**. Estudiar una función AutoLISP implica conocer cómo opera y qué valor devuelve.

Los códigos llevan una estructura anidada; por ejemplo, considérense estos dos formatos: (+ *sum1 sum2 ...*) es la sintaxis de la función de suma y (setq *var1 expr1*) es la función de asignación, que valora la variable *var1* con el resultado de la expresión *expr1*. Entonces, (+ 3 4) efectúa la suma y devuelve al exterior el valor 7; (setq *a* (+ 3 4)) es evaluada mediante la regla clásica: desde el paréntesis más interno hacia los más externos; de ese modo, se evalúa la suma y el valor 7 es expulsado al exterior. Ese valor expulsado hace de segundo argumento en la función setq, de modo que la variable *a* pasa a tomar el valor 7 (y el tipo entero). A su vez, la función setq devuelve al exterior el valor evaluado, 7, que al no formar parte de una nueva función más externa, se pierde.

Los procedimientos tienen el formato (**DEFUN nombre (argumentos) func1 func2 ...**), donde func1, func2, etc son funciones de biblioteca (anidadas o no) u otros procedimientos: también responden a una estructura de lista. Al final de este capítulo se presentan algunos códigos ilustrativos (ver el apartado 11.4). Incluso, como se verá en el capítulo 13, los registros asociados a una entidad se articulan en forma de lista; esta característica da un sentido muy uniforme a este lenguaje de programación. En el apartado 11.2 se irán describiendo las funciones básicas junto con algunos ejemplos y observaciones; en el apartado 11.4 se estudian técnicas básicas de aplicación práctica.

¹ LISP es una contracción de la expresión LISt Processor: Procesador de Listas.