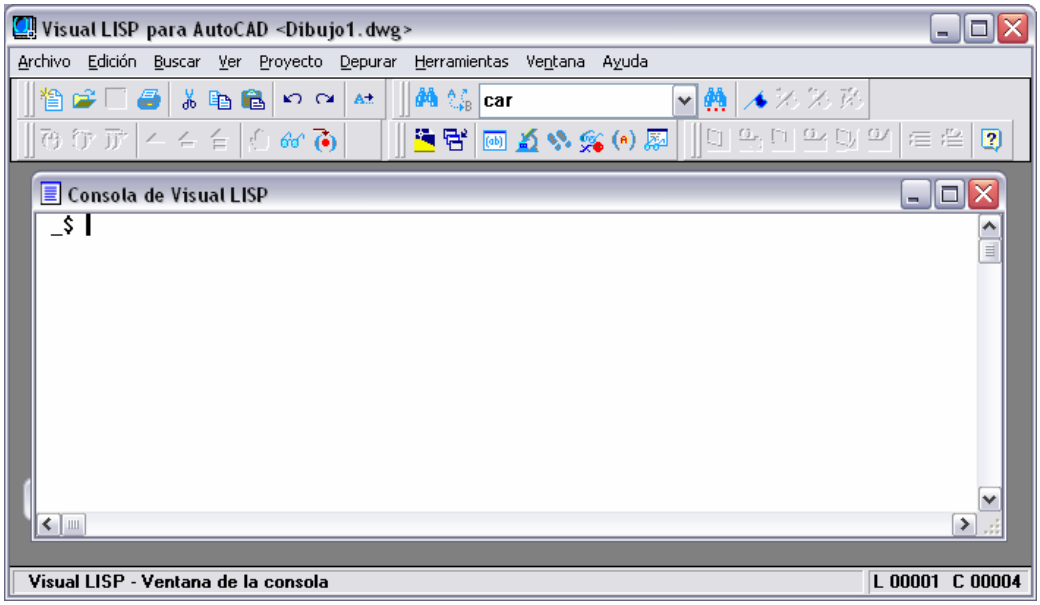
BREVE DESCRIPCIÓN DEL ENTORNO DE PROGRAMACIÓN DE VISUAL LISP

1. INTRODUCCIÓN

El entorno de desarrollo de Visual Lisp es accesible desde Autocad de dos maneras distintas:

- ✓ Desde el menú *herramientas* → *AutoLISP* → *Editor de Visual Lisp*
- ✓ Escribiendo el comando *VLISP* o el comando *VLIDE* en la línea de comandos



Cuando iniciamos VLISP observamos los siguientes componentes:

- ✓ Menú: hay una línea de menú en la parte superior de la ventana. Si la exploramos podemos encontrar diversas herramientas, que veremos en apartados posteriores.
- ✓ Barras de herramientas. Existen cinco barras distintas: debug, edit, find, Inspect y Run. Con las barras de herramientas se pueden ejecutar la mayoría de comandos existentes en la barra de menús. Cuando posicionamos el ratón encima de cualquier herramienta veremos aparecer una breve descripción de la misma en la barra de estado en la parte inferior de la ventana de VLISP.
- ✓ Ventana de Consola. Es una ventana en la que podremos ejecutar comandos VLISP de modo similar a la ventana de comandos de AutoCAD. La vemos de más detenidamente en el apartado 2
- ✓ Barra de estado: va presentando diversas informaciones según lo que estemos haciendo en VLISP
- ✓ Ventana de rastreo. Muestra mensajes adicionales sobre la versión de VLISP y errores varios que van apareciendo.

2. VENTANA DE CONSOLA

La consola VLISP es en muchos aspectos similar a la ventana de comandos de AutoCAD, pero proporciona muchas más utilidades y su funcionamiento, aunque similar, difiere en varios aspectos. La consola es una herramienta de mucha mayor potencia, y existen diferencias en el modo de operación.

Por ejemplo, para conocer el valor asociado a un símbolo, en la consola simplemente escribiremos dicho símbolo y al pulsar Enter obtendremos su resultado. En AutoCAD, en cambio, deberemos preceder a este símbolo de un signo de admiración (!)

Las prestaciones más importantes de la consola son:

- ✓ Evaluar expresiones LISP y mostrar el resultado que devuelven
- ✓ Introducir expresiones LISP en líneas múltiples, utilizando CTRL+ENTER como salto de línea.
- ✓ Cortar y pegar texto
- ✓ Recuperar expresiones tecleadas anteriormente pulsando la tecla TAB. A medida que vamos pulsando va avanzando por el histórico de instrucciones. Es posible avanzar en el sentido contrario pulsando SHIFT+TAB
- ✓ Al pulsar ESC se cancela la instrucción actual. Si se pulsa SHIFT + ESC aparece una nueva línea de consola, pero sin perder lo escrito anteriormente.
- ✓ Menú contextual al pulsar el botón derecho del ratón

Un detalle a tener en cuenta también es la detección, por parte de la consola, de lo que estamos escribiendo, de modo que cambia el color de la letra según lo que escribamos.

- Funciones de VLISP y símbolos protegidos → color azul
- Cadenas de caracteres → color magenta
- Números enteros → color verde
- Números reales → color verde mar
- Comentarios → color magenta sobre fondo gris
- Paréntesis → color rojo
- Resto del texto → color negro

3. EDITOR DE TEXTO DE VISUAL LISP

El editor de texto VLISP es el componente central del entorno de desarrollo. Algunas de las características del editor son:

- ✓ Codificación de colores: al igual que en la consola, el editor de texto distingue los símbolos protegidos, operadores y demás elementos del lenguaje.

- ✓ Formateo de texto. Ofrece distintos tipos de formato que nos harán el código más facil de leer.
- ✓ Comprobación de los paréntesis. El editor es capaz de encontrar hasta dónde llegan los paréntesis.
- ✓ Ejecución de expresiones LISP. Se pueden testear expresiones sin salir del editor
- ✓ Búsqueda de palabras
- ✓ Comprobación de la sintaxis del código

Para crear un nuevo archivo LISP, iremos al menú *Archivo → Nuevo Archivo*. Aparecerá una nueva ventana de texto vacía. También podemos pulsar el icono correspondiente en la barra de herramientas  .

A medida que vayamos escribiendo código, veremos que el editor lo formatea de modo automático a medida que vamos añadiendo nuevas líneas. De todos modos, en caso de que añadamos código sin formato, lo podemos formatear desde el menú *Herramientas → Formatear Código en Editor* o con el comando  . También se podrá formatear una selección de código desde el menú *Herramientas → Formatear selección* o con el comando  .

El editor proporciona también las típicas herramientas de edición como cortar, pegar, deshacer, rehacer, etc, accesibles desde el menú correspondiente o desde la barra de herramientas.

Y finalmente tenemos la herramienta de comprobación de errores de sintaxis, que permite detectar los siguientes errores:

- ✓ Número incorrecto de argumentos
- ✓ Nombre inválido de variable pasado a una función
- ✓ Sintaxis incorrecta en llamada a función

Para ejecutar el comprobador de código, vamos al menú *Herramientas → Comprobar Texto en Editor* o pulsamos el icono  . Si solo queremos comprobar parte del código, seleccionaremos dicho código en la ventana de texto y ejecutaremos *Herramientas → Comprobarr Selección,* o pulsaremos el icono  . Si existe algún error, este se mostrará en una ventana llamada Ventana de Salida. Y si hacemos doble clic sobre el error, éste se resaltará en el código

4. CARGAR, Y EJECUTAR PROGRAMAS

Una vez que ya tenemos una función LISP escrita, necesitamos "cargarla" para poder ejecutarla. Esto se hace desde el menú *Herramientas → Cargar Texto en Editor* o pulsando

el icono  o si solo queremos cargar una selección, desde el menú *Herramientas* → *Cargar selección en Editor* o pulsando el icono  . Una vez hecho esto no hay más que llamar a la función desde la consola o desde la ventana de comandos de Autocad.

5. DEPURACIÓN DE PROGRAMAS

Para depurar un programa, hay que ejecutar en forma de rastreo, de modo que podamos ver el valor de las variables durante la ejecución, y cómo se evalúan. Esto es útil porque a veces, cuando ejecutamos un programa da resultados erróneos y no sabemos por qué. VLISP ofrece las siguientes herramientas de depuración:

- ✓ Ejecuciones de código paso a paso
- ✓ Rastreo de los valores de las variables durante la ejecución.
- ✓ Rastreo de la secuencia de ejecución.
- ✓ Puntos de interrupción
- ✓ Inspección de la pila

Todas estas herramientas vienen descritas en el manual de ayuda de VLISP. En este documento sólo nos vamos a centrar en el funcionamiento de los puntos de interrupción y la ejecución de código paso a paso.

Los puntos de interrupción nos permiten marcar una posición en el código en donde queremos que se interrumpa la ejecución. Se deben poner antes o después de una expresión entre paréntesis. Para crear un punto de interrupción:

- ✓ Mover el cursor a la posición en dónde queremos crear el punto de interrupción.
- ✓ Pulsar el botón  o pulsar F9 para fijar el punto de interrupción. También se puede acceder desde el menú *Depurar* → *Act/Des Punto de Interrupción*. Si el cursor está posicionado en un lugar incorrecto, como en medio de una expresión, VLISP moverá el cursor al paréntesis más cercano, y preguntará si se quiere posicionar en él el punto de interrupción.
- ✓ Para borrar un punto de interrupción se sigue la misma secuencia. Para borrarlos todos, iremos al menú *Depurar* → *Borrar todos los puntos de interrupción.*

Una vez fijados los puntos de interrupción deseados, ejecutaremos el programa y veremos que la ejecución se detiene en el punto deseado. En esta situación podemos evaluar los valores de las variables en ese punto. Si queremos continuar la ejecución del programa tenemos varias opciones:

- ✓ Continuar ejecutando paso a paso, pulsando el botón  o la tecla F8

- ✓ Continuar ejecutando la siguiente expresión, pulsando el botón  o mayúsculas + F8
- ✓ Continuar ejecutando hasta un nuevo punto de interrupción o hasta el final, pulsando el botón o CTRL + F8
- ✓ Salir del bucle para ir al inmediato superior pulsando el botón  o CTRL +Q
- ✓ Parar la ejecución pulsando  o CTRL + R

De todos modos, el modo más facil de aprender el uso de estas herramientas es utilizándolas en algún ejemplo.

El entorno VLISP proporciona muchas más herramientas, cuya descripción aparece de forma extensa en la ayuda que proporciona el entorno