

Trabajo dibujo de sala CAD en Autolisp

Computación gráfica

David Aguirre Rocha

Profesor: Miguel Ángel Baquero Cortés



Facultad de ingeniería
Departamento de ingeniería mecánica y mecatrónica
Universidad Nacional de Colombia
Bogotá, Colombia
Octubre 30 2022

1 Descripción del trabajo

Se desarrolló una rutina en Autolisp que se encarga de leer información de distintos bloques para construir un dibujo aproximado de la sala CAD de la Universidad Nacional de Colombia. Esta información puede presentarse en 3 formatos:

- Archivo de texto con información separada por comas.
- Archivo de texto con información separada por espacios.
- Archivo de Excel.

La información de estos archivos es de posición y de atributos definidos por el usuario. La información de los dibujos está en archivos .dwg por separado dibujados manualmente una vez. En el presente documento se muestra el desarrollo de esta rutina, los archivos que la componen y cómo correrla.

2 Dibujo manual de Sala CAD

2.1 Zona de trabajo individual

En primer lugar se definió una mesa de trabajo individual constituida por los siguientes elementos sobre una mesa genérica:

- Cuaderno
- Monitor
- Smartphone
- Mouse
- Teclado
- Silla

La mesa de trabajo propuesta inicialmente se dibujó de forma manual y se aprecia en la figura 1.

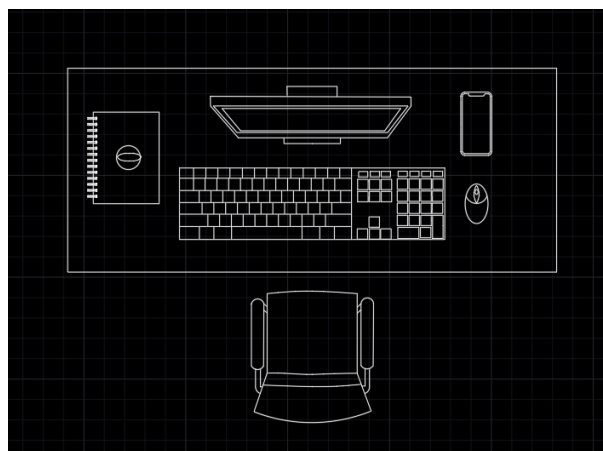


Figura 1: Zona de trabajo individual

2.2 Sala CAD

Ahora bien se estableció que para la sala CAD hay 4 bloques donde cada uno cuenta con 4 estaciones individuales como se aprecia en la figura 2.



Figura 2: Sala CAD dibujada manualmente

3 Definición de bloques y atributos

Ahora se asignan como bloques a cada uno de los elementos de la figura 1 sin incluir la mesa pues se considera esta genérica. Para cada bloque se definieron 5 atributos. Estos atributos no son reales, se toman datos aproximados para fines del ejercicio.

1. Color: Color del elemento.
2. Fecha: Año de manufactura del elemento.
3. Marca: Empresa fabricante del elemento.
4. Precio: Precio de compra del elemento.
5. Referencia: Referencia del elemento por la marca.

En la tabla 1 se muestran los valores asignados para cada uno de los atributos de cada elemento.

ELEMENTO	COLOR	PRECIO	MARCA	REFERENCIA	FECHA
CUADERNO	NEGRO	30000	Scribe	100 hojas	2019
MONITOR	ROJO	1500000	SAMSUNG	AMBX	2021
PHONE	ROJO	4000000	APPLE	IPHONE 14	2022
MOUSE	VERDE	200000	LOGITECH	SS45	2018
TECLADO	AZUL	300000	RAZER	MJ33	2021
SILLA	VERDE	300000	HOME & OFF	AMB55	2018

Tabla 1: Atributos de cada elemento

4 Extracción de información de cada bloque

Después de definir los bloques, se hizo uso del commando ATTEXT de autoCAD para extraer la información de cada bloque, sin embargo en primer lugar es necesario crear

una plantilla que establezca cuáles son los parámetros a extraer. En el siguiente [enlace](#) se explica de forma detallada cómo construir esta plantilla.

La plantilla construida fue la mostrada en la figura 3.

```
BL:NAME C020000
BL:X N004003
BL:Y N004003
BL:Z N004003
COLOR C020000
PRECIO N008000
MARCA C020000
REFERENCIA C020000
FECHA N005000
```

Figura 3: Plantilla para ejecución de ATTEXT.

De esta manera se extrae de cada bloque: Su nombre, posición x , posición y y posición z en el dibujo de la figura 1 y los atributos mencionados anteriormente. Los comandos con prefijo 'C' y 'N' es para indicar el número de caracteres y cifras numéricas de cada dato respectivamente. Haciendo uso de esta plantilla se exportaron los datos en 3 formatos.

```
'CUADERNO',224.,167.,0.00,'NEGRO', 30000,'Scribe','100 hojas', 2019
'MONITOR',600.,315.,0.00,'ROJO', 1500000,'SAMSUNG','AMBX', 2021
'PHONE',1003,286.,0.00,'ROJO', 4000000,'APPLE','IPHONE 14', 2022
'MOUSE',975.,167.,0.00,'VERDE', 200000,'LOGITECH','SS45', 2018
'TECLADO',600.,255.,0.00,'AZUL', 300000,'RAZER','MJ33', 2021
'SILLA',600.,-46.,0.00,'VERDE', 300000,'HOME & OFFICE','AMB55', 2018
```

Figura 4: Archivo de texto separado por comas.

CUADERNO	224.167.0.00	NEGRO	30000	Scribe	100 hojas	2019
MONITOR	600.315.0.00	ROJO	1500000	SAMSUNG	AMBX	2021
PHONE	1003286.0.00	ROJO	4000000	APPLE	IPHONE 14	2022
MOUSE	975.167.0.00	VERDE	200000	LOGITECH	SS45	2018
TECLADO	600.255.0.00	AZUL	300000	RAZER	MJ33	2021
SILLA	600.-46.0.00	VERDE	300000	HOME & OFFICE	AMB55	2018

Figura 5: Archivo de texto separado por espacio.

	A	B	C	D	E	F	G	H	I
1	CUADERNO	224.	167.	0.00	NEGRO	30000	Scribe	100 hojas	2019
2	MONITOR	600.	315.	0.00	ROJO	1500000	SAMSUNG	AMBX	2021
3	PHONE	1003	286.	0.00	ROJO	4000000	APPLE	IPHONE 14	2022
4	MOUSE	975.	167.	0.00	VERDE	200000	LOGITECH	SS45	2018
5	TECLADO	600.	255.	0.00	AZUL	300000	RAZER	MJ33	2021
6	SILLA	600.	-46.	0.00	VERDE	300000	HOME & OFFICE	AMB55	2018

Figura 6: Archivo de excel.

Para los archivos separados por comas y por espacios se obtienen directamente al hacer uso de la función ATTEXT con su respectiva plantilla. Para el archivo de Excel, el programa importó adecuadamente el archivo separado por comas y creó tales celdas de excel.

5 Código desarrollado

Para el código desarrollado se dividió en las siguientes funciones creadas por mi por facilidad de programación y de entendimiento:

1. C:LEERPLANT (). Función para leer la información descrita en la plantilla.
2. C:ASC (). Función para leer el archivo de texto separado por comas y dibujar la sala CAD.
3. C:ASE (). Función para leer el archivo de texto separado por espacios y dibujar la sala CAD.
4. C:ASXL (). Función para leer el archivo de excel y dibujar la sala CAD.
5. C:SALACAD (). Función que pregunta al usuario qué tipo de lectura quiere realizar y llama a la función correspondiente (ASC, ASE o ASXL) para dibujar la sala CAD.

Antes de explicar el funcionamiento de las funciones se debe aclarar que para insertar los bloques en la rutina se hace mediante archivos .dwg donde está definido cada elemento individual con sus atributos y se inserta entonces cada archivo .dwg como bloque independiente.

5.1 C:LEERPLANT ()

```
1 ;Función para leer información de la plantilla.
2 (defun C:LEERPLANT ()
3   (setq plantilla (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\Plantilla.txt") "r"))
4   (setq atributostotal 1)
5
6   (setq renfut (read-line plantilla))
7
8
9   (while (/= (/= renfut nil) nil)
10    (setq renfut (read-line plantilla))
11    (setq atributostotal (+ atributostotal 1))
12  )
13  (setq plantilla (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\Plantilla.txt") "r"))
14  (setq renglon 1)
15  (setq renaact "")
16  (setq ch 1)
17
18  (setq renfut (read-line plantilla))
19  (setq infoplantilla (vlax-make-safearray vlax-vbString (cons 1 (- atributostotal 1)) (cons 1 4)))
20
21  (while (/= (/= renfut nil) nil)
22    (setq renaact renfut)
23    (repeat (strlen renaact)
24      (if (= (substr renaact ch 1) " ")
25        (progn
26          (setq infoplant (substr renaact (+ ch 1)))
27          (vlax-safearray-put-element infoplantilla renglon 1 (substr renaact 1 (- ch 1)))
28          (vlax-safearray-put-element infoplantilla renglon 2 (substr infoplant 1 1))
29          (vlax-safearray-put-element infoplantilla renglon 3 (substr infoplant 2 3))
30          (vlax-safearray-put-element infoplantilla renglon 4 (substr infoplant 5 3))
31        )
32      )
33    (setq ch (+ ch 1))
34  )
35  (setq renglon (+ renglon 1))
36  (setq ch 1)
37  (setq renfut (read-line plantilla))
38 )
39 )
40
```

Figura 7: Función C:LEERPLANT ()

En esta función se extrae la información de la plantilla, es decir para cada atributo se extrae el nombre del atributo, si se tiene un valor numérico **N** o de texto **C**, también la longitud que tiene cada uno de estos dado por los primeros 3 dígitos. También se extrae el número total de atributos que se tiene para cada elemento. Esta información es necesaria para el funcionamiento de las demás funciones definidas.

5.2 C:ASC ()

```

1 ;Función para leer archivo separado por comas y dibujar sala CAD.
2 (defun C:ASC ()
3   (setq plantilla (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\ASC.txt") "r"))
4   (setq elementos 1)
5   (setq renfut (read-line plantilla))
6
7   (while (/= renfut nil))
8     (setq renfut (read-line plantilla))
9     (setq elementos (+ elementos 1))
10  )
11
12  (setq mesacoma (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\ASC.txt") "r"))
13  (setq nf -1)
14  (setq renact "")
15  (setq ch 1)
16  (setq ci 1)
17  (setq renglon 1)
18  (setq atributos 1)
19  (setq infocomas (vlax-make-safearray vlax-vbString (cons 1 (- elementos 1)) (cons 1 (- atributostotal 1))))
20
21  (setq renfut (read-line mesacoma))
22  (while (/= renfut nil))
23    (setq renact renfut)
24    (repeat (strlen renact)
25      (if (= (substr renact ch 1) ",")
26        (progn
27          (vlax-safearray-put-element infocomas renglon atributos (substr renact ci (- ch ci)))
28          (setq ci (+ ch 1))
29          (setq atributos (+ atributos 1))
30          (setq ch (+ ch 1))
31        )
32        (setq ch (+ ch 1))
33      )
34      (if (= ch (strlen renact))
35        (vlax-safearray-put-element infocomas renglon atributos (substr renact ci (+ (- ch ci) 1)))
36      )
37    )
38  (setq atributos 1)
39  (setq ci 1)
40  (setq renglon (+ renglon 1))
41  (setq ch 1)
42  (setq nf (+ nf 1))
43  (setq renfut (read-line mesacoma))
44  )
45  ;Dibujo de la sala CAD.
46  (setq nombres (vlax-make-safearray vlax-vbString '(1 . 6)))
47
48  (vlax-safearray-put-element nombres 1 "CuadernoE.dwg")
49  (vlax-safearray-put-element nombres 2 "MonitorE.dwg")
50  (vlax-safearray-put-element nombres 3 "Phonete.dwg")
51  (vlax-safearray-put-element nombres 4 "MouseE.dwg")
52  (vlax-safearray-put-element nombres 5 "TecladoE.dwg")
53  (vlax-safearray-put-element nombres 6 "Sillaf.dwg")
54
55  (setq j 0)
56  (setq k 500)
57  (repeat 4
58    (setq i 1)
59
60    (repeat (- elementos 1)
61      (setq cordx (vlax-safearray-get-element infocomas i 2))
62      (setq cordy (vlax-safearray-get-element infocomas i 3))
63      (setq cordz (vlax-safearray-get-element infocomas i 4))
64      (setq cordxn (+ (read cordx) j))
65      (setq cordx (rtos cordxn 2 6))
66      (command "_insert" (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\") (vlax-safearray-get-element nombres i)) (strcat cordx ", " cordy) "1" "1" "")
67      (setq i (+ i 1))
68    )
69    (setq mesal (strcat (rtos j 2 6) ", " (rtos 0 2 6)))
70    (setq mesaf (strcat (rtos (+ j 1200) 2 6) ", " (rtos k 2 6)))
71    (command "_rectang" mesal mesaf "")
72    (setq j (+ j 1200))
73  )
74  (setq esp1 (strcat (rtos (+ j 400) 2 6) ", " "0"))
75  (setq esp2 (strcat (rtos (+ j 400) 2 6) ", " "10"))
76  (command "_mirror" (ssget "_A" "" "0,500" "10,500" "No")
77    (command "_mirror" (ssget "_A" "" "0,500" "10,500" "No")
78      (command "_mirror" (ssget "_A" "" "0,1500" "10,1500" "No")
79        (command "_mirror" (ssget "_A" "" "0,1500" "10,1500" "No")
80          )
81      )
82    )
83  )

```

Figura 8: Función C:ASC ()

En esta función se extrae la información del archivo de texto separado por comas, para ello lee la información renglón por renglón y en cada renglón lee caracter por caracter buscando una coma, cuando el programa encuentra la coma almacena todos los caracteres anteriores

y los guarda en un arreglo donde las el número de filas es el renglón leído y el número de columnas es el que distingue el valor de cada atributo.

Luego para el dibujo primero se definen los nombres de los archivos .dwg, se extrae la información del archivo separado por comas almacenada en el arreglo mencionado, con esta información se dibuja una zona de trabajo individual (coordenadas almacenadas) y a partir de esta se realizan múltiples reflexiones en función de construir la sala CAD constituida por varias zonas de trabajo individual.

5.3 C:ASE ()

```

1 ;Función para leer archivo separado por espacios y dibujar sala CAD.
2 (defun C:ASE ()
3   (setq archivosp (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\ASE.txt") "r"))
4   (setq elementos 1)
5
6   (setq renfut (read-line archivosp))
7
8
9   (while (/= renfut nil) nil)
10  (setq renfut (read-line archivosp))
11  (setq elementos (+ elementos 1))
12
13
14  (setq mesaespacio (open (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\ASE.txt") "r"))
15  (setq nf -1)
16  (setq renact "")
17  (setq ch 1)
18  (setq ci 1)
19  (setq renglon 1)
20  (setq atributos 1)
21  (setq infoespacio (vlax-make-safearray vlax-vbString (cons 1 (- elementos 1)) (cons 1 (- atributostotal 1))))
22
23  (setq renfut (read-line mesaespacio))
24  (while (/= renfut nil) nil)
25  (setq renact renfut)
26  (repeat (- atributostotal 1)
27    (setq longat (read (vlax-safearray-get-element infoespacio atributos 3)))
28    (if (= "C" (vlax-safearray-get-element infoespacio atributos 2))
29      (vlax-safearray-put-element infoespacio renglon atributos (vl-string-right-trim " " (substr renact ci longat)))
30      (vlax-safearray-put-element infoespacio renglon atributos (vl-string-left-trim " " (substr renact ci longat)))
31    )
32    (setq ci (+ ci longat))
33    (setq atributos (+ atributos 1))
34  )
35  (setq atributos 1)
36  (setq ci 1)
37  (setq renglon (+ renglon 1))
38  (setq ch 1)
39  (setq renfut (read-line mesaespacio))
40
41
42  (setq nombres (vlax-make-safearray vlax-vbString '(1 . 6)))
43
44  (vlax-safearray-put-element nombres 1 "CuadernoE.dwg")
45  (vlax-safearray-put-element nombres 2 "MonitorE.dwg")
46  (vlax-safearray-put-element nombres 3 "PhoneE.dwg")
47  (vlax-safearray-put-element nombres 4 "MouseE.dwg")
48  (vlax-safearray-put-element nombres 5 "TecladoE.dwg")
49  (vlax-safearray-put-element nombres 6 "SillaE.dwg")
50
51  (setq j 0)
52  (setq k 500)
53  (repeat 4
54    (setq i 1)
55    (repeat (- elementos 1)
56      (setq cordx (vlax-safearray-get-element infoespacio i 2))
57      (setq cordy (vlax-safearray-get-element infoespacio i 3))
58      (setq cordz (vlax-safearray-get-element infoespacio i 4))
59      (setq cordxn (+ (read cordx) j))
60      (setq cordx (rtos cordxn 2 6))
61      (command "_insert" (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\") (vlax-safearray-get-element nombres i) (strcat cordx ", " cordy) "1" "1" "")
62      (setq i (+ i 1))
63    )
64    (setq mesai (strcat (rtos j 2 6) ", " (rtos 0 2 6)))
65    (setq mesaf (strcat (rtos (+ j 1200) 2 6) ", " (rtos k 2 6)))
66    (command "_rectang" mesai mesaf "")
67    (setq j (+ j 1200))
68  )
69  (setq esp1 (strcat (rtos (+ j 400) 2 6) ", " "0"))
70  (setq esp2 (strcat (rtos (+ j 400) 2 6) ", " "10"))
71  (command "_mirror" (ssget "_A") "" "0,500" "10,500" "No")
72  (command "_mirror" (ssget "_A") "" esp1 esp2 "No")
73  (command "_mirror" (ssget "_A") "" "0,1500" "10,1500" "No")
74 )

```

Figura 9: Función C:ASE ()

En esta función se lee renglón por renglón del archivo de texto separado por espacios y teniendo definidas la longitud de cada atributo (extraído de la plantilla) es posible almacenar cada atributo y eliminar los espacios restantes. Estos datos se almacenan en

un arreglo donde las el número de filas es el renglón leído y el número de columnas es el que distingue el valor de cada atributo.

Luego para el dibujo primero se definen los nombres de los archivos .dwg, se extrae la información del archivo separado por espacios almacenada en el arreglo mencionado, con esta información se dibuja una zona de trabajo individual (coordenadas almacenadas) y a partir de esta se realizan múltiples reflexiones en función de construir la sala CAD constituida por varias zonas de trabajo individual.

5.4 C:ASXL

```
1 ;Función para leer archivo de excel y dibujar sala CAD.
2 (defun C:ASXL ()
3   (C:GetCells)
4   (setq atributostotal elementos)
5   (setq elementos 7)
6   (setq infoexcel (vlax-make-safearray vlax-vbString (cons 1 (- elementos 1)) (cons 1 (- atributostotal 1))))
7
8   (setq cellrange (strcat "A1:" (chr (+ 63 atributostotal)) (rtos (- elementos 1) 2 6)))
9
10  (setq infoexp (getCellFunction (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\AXL.xlsx") "Hoja1" cellrange))
11
12  (setq renglon 1)
13  (setq atributos 1)
14  (setq cont 0)
15  (repeat (- elementos 1)
16    (vlax-safearray-put-element infoexcel renglon atributos (nth cont infoexp))
17    (setq atributos (+ atributos 1))
18    (setq cont (+ cont 1)))
19  )
20  (setq atributos 1)
21  (setq renglon (+ renglon 1))
22  )
23  )
24
25  (setq nombres (vlax-make-safearray vlax-vbString '(1 . 6)))
26
27  (vlax-safearray-put-element nombres 1 "CuadernoE.dwg")
28  (vlax-safearray-put-element nombres 2 "MonitorE.dwg")
29  (vlax-safearray-put-element nombres 3 "PhoneE.dwg")
30  (vlax-safearray-put-element nombres 4 "MouseE.dwg")
31  (vlax-safearray-put-element nombres 5 "TecladoE.dwg")
32  (vlax-safearray-put-element nombres 6 "SillaE.dwg")
33
34  (setq j 0)
35  (setq k 500)
36  (repeat 4
37    (setq i 1)
38    (repeat (- elementos 1)
39      (setq cordx (vlax-safearray-get-element infoexcel i 2))
40      (setq cordy (vlax-safearray-get-element infoexcel i 3))
41      (setq cordz (vlax-safearray-get-element infoexcel i 4))
42      (setq cordxn (+ (read cordx) j))
43      (setq cordz (rtos cordxn 2 6))
44      (command "_insert" (strcat (findfile "Archivos_salaCAD_David_Aguirre") "\\") (vlax-safearray-get-element nombres i) (strcat cordx "," cordy) "1" "1" "")
45      (setq i (+ i 1)))
46    )
47    (setq mesai (strcat (rtos j 2 6) "," (rtos 0 2 6)))
48    (setq mesaf (strcat (rtos (+ j 1200) 2 6) "," (rtos k 2 6)))
49    (command "_rectang" mesai mesaf "")
50    (setq j (+ j 1200))
51  )
52  (setq esp1 (strcat (rtos (+ j 400) 2 6) "," "9"))
53  (setq esp2 (strcat (rtos (+ j 400) 2 6) "," "10"))
54  (command "_mirror" (ssget "_A" "" "0,500" "10,500" "No")
55    (command "_mirror" (ssget "_A" "" esp1 esp2 "No")
56      (command "_mirror" (ssget "_A" "" "0,1500" "10,1500" "No")
57        )
58    )
59  )
```

Figura 10: Función C:ASXL ()

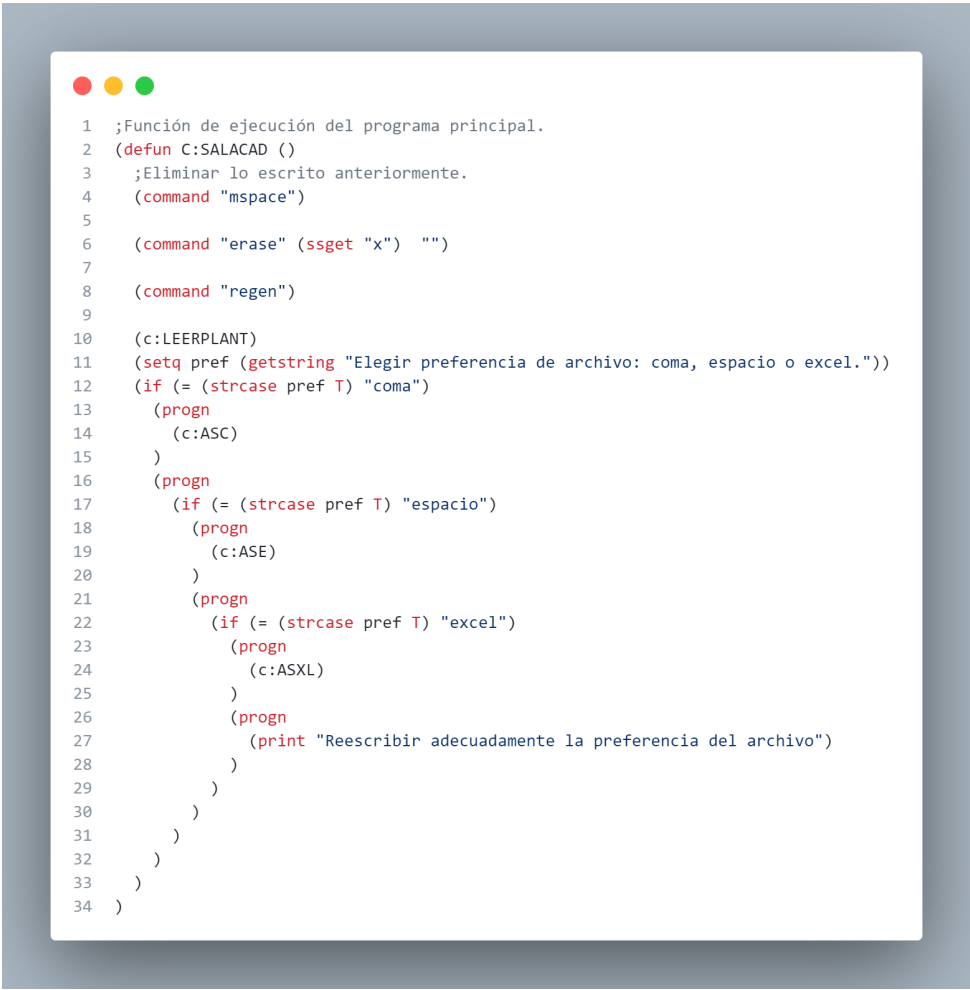
Para esta función se hizo uso de una sub-función llamada C:GetCells encontrada en internet que recibe como entrada un rango de celdas y un archivo de excel para extraer esa información en formato de lista en LISP, se estudió el código a profundidad y hace uso de diversas funciones VL con programación orientada a objetos. En [este enlace](#) se explica a profundidad el funcionamiento de esta función y se da crédito al autor pues en mi opinión implementarla en futuros códigos es sumamente ventajoso.

Ahora bien, la función C:ASXL si es de mi autoría y esta función se encarga de transformar la información de plantilla para identificar el tamaño de la cuadrícula del excel, luego llama

a la función GetCells con la dimensión adecuada de las celdas y se extrae la información en forma de lista, luego sabiendo el número de atributos y de elementos a estudiar se cambia el formato de los datos de un arreglo unidimensional a un arreglo bidimensional como en las funciones anteriores, es decir, al final estos datos se almacenan en un arreglo donde las el número de filas es el renglón leído y el número de columnas es el que distingue el valor de cada atributo.

Luego para el dibujo primero se definen los nombres de los archivos .dwg, se extrae la información del archivo de excel almacenada en el arreglo mencionado, con esta información se dibuja una zona de trabajo individual (coordenadas almacenadas) y a partir de esta se realizan múltiples reflexiones en función de construir la sala CAD constituída por varias zonas de trabajo individual. Igualmente, se adjunta como Anexo una imagen del código completo.

5.5 Código de ensamble final Sala CAD.



```
1 ;Función de ejecución del programa principal.
2 (defun C:SALACAD ()
3   ;Eliminar lo escrito anteriormente.
4   (command "mspace")
5
6   (command "erase" (ssget "x") "")
7
8   (command "regen")
9
10  (c:LEERPLANT)
11  (setq pref (getstring "Elegir preferencia de archivo: coma, espacio o excel.))
12  (if (= (strcase pref T) "coma")
13    (progn
14      (c:ASC)
15    )
16    (progn
17      (if (= (strcase pref T) "espacio")
18        (progn
19          (c:ASE)
20        )
21        (progn
22          (if (= (strcase pref T) "excel")
23            (progn
24              (c:ASXL)
25            )
26            (progn
27              (print "Reescribir adecuadamente la preferencia del archivo")
28            )
29          )
30        )
31      )
32    )
33  )
34 )
```

Figura 11: Rutina final del programa.

En este programa se define que el usuario ingrese un comando de texto en función del tipo de archivo de construcción que desee ver al final. Para todos los archivos debería ser exactamente el mismo dibujo representado al final a pesar de que estén codificados diferente por la naturaleza del archivo. Sin embargo se dejará las fuentes principales que se trabajaron. Si la entrada es distinta a las opciones solicitadas el programa informa que debe ingresar un valor correcto.

6 ¿Cómo cargar y correr el programa?

1. En AutoCAD, la barra inferior derecha debe verse de la siguiente manera de tal forma que se desactiven los snaps y permitan que el programa dibuje libremente por más de que se acerque a otras líneas.



Figura 12: Barra de configuración AutoCAD.

2. Debe colocarse la carpeta adjunta con nombre 'Archivos_salaCAD_David_Aguirre' en la carpeta de AutoCAD 2023 donde se encuentre instalado AutoCAD o donde se esté trabajando para no tener que recurrir a rutas específicas del computador. Esta carpeta debe verse como en la figura 13.

Este equipo > Datos (D:) > AutoCAD 2023 > Archivos_salaCAD_David_Aguirre				
Nombre	Fecha de modificación	Tipo	Tamaño	
ASC	23/10/2022 10:00 a. m.	Documento de tex...	1 KB	
ASE	26/10/2022 10:23 a. m.	Documento de tex...	1 KB	
AXL	28/10/2022 9:20 a. m.	Hoja de cálculo d...	11 KB	
CuadernoE	22/10/2022 6:01 p. m.	Archivo DWG	37 KB	
MonitorE	22/10/2022 6:00 p. m.	Archivo DWG	34 KB	
MouseE	22/10/2022 6:03 p. m.	Archivo DWG	35 KB	
PhoneE	22/10/2022 6:01 p. m.	Archivo DWG	35 KB	
Plantilla	23/10/2022 11:02 a. m.	Documento de tex...	1 KB	
SillaE	23/10/2022 9:59 a. m.	Archivo DWG	35 KB	
TecladoE	22/10/2022 6:02 p. m.	Archivo DWG	40 KB	

Figura 13: Archivos de carpeta 'Archivos_salaCAD_David_Aguirre'.

3. Correr el programa adjunto 'SalaCAD_David_Aguirre.lsp'.
4. Escribir en la barra de comandos de AutoCAD 'salacad'.
5. Escribir 'coma', 'espacio' o 'excel en función del tipo de archivo que se quiera leer.
6. Si la entrada es diferente a 'coma', 'espacio' o 'excel' el programa escribirá una alerta indicando que debe volverse a escribir el comando adecuadamente.
7. Después de realizar un dibujo, si desea realizar otro, repetir desde el paso 3.

7 Imagen de sala CAD

7.1 Sala CAD de archivo separado por comas.

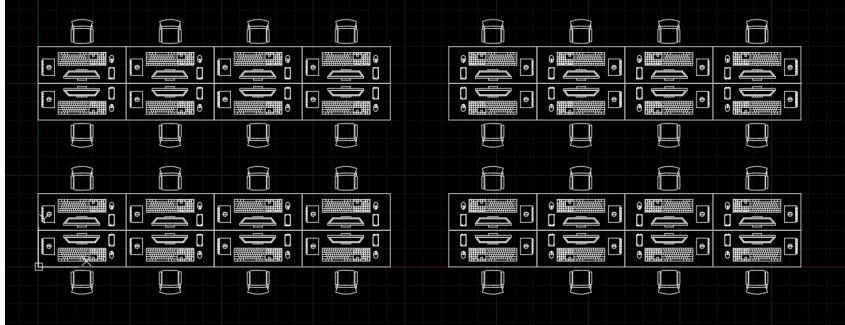


Figura 14: Dibujo de lectura del archivo separado por comas.

7.2 Sala CAD de archivo separado por espacios.



Figura 15: Dibujo de lectura del archivo separado por espacios.

7.3 Sala CAD de archivo excel.

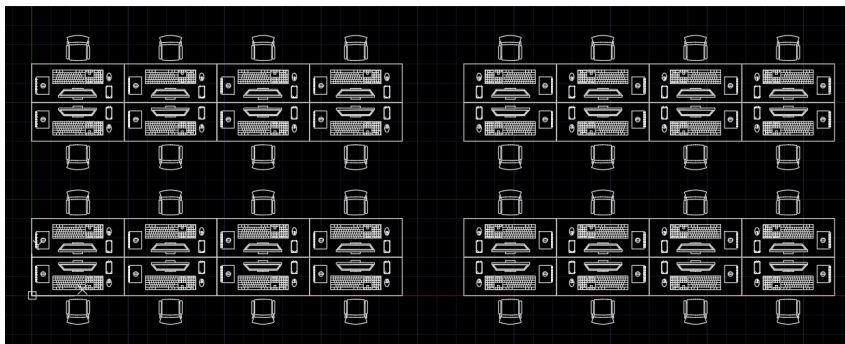


Figura 16: Dibujo de lectura del archivo de excel.

En las figuras 14, 15, 16 es apreciable que los dibujos son prácticamente idénticos a pesar de que el programa lee la información de los archivos de distinta manera por lo que estaría

funcionando correctamente. De todas formas es posible comprobar la rutina en AutoCAD.

8 Conclusiones del trabajo

- Los 3 programas, lectura de archivo de texto separado por comas (C:ASC), lectura de archivo de texto separado por espacios (C:ASE) y lectura de archivo excel (C:ASXL) dibujan correctamente la sala CAD según lo deseado inicialmente.
- La lectura de archivos de texto separados por coma o por espacio o archivo de excel, permite automatizar tareas sencillas de AutoCAD en diversos contextos.
- Las funciones con prefijo vl permiten la manipulación de archivos de forma sencilla.
- Almacenar información de distintos bloques en archivos de texto permite el ahorro considerable de memoria.
- Definir atributos personalizados para bloques permite un mejor manejo de la información en distintos contextos aplicables en autoCAD.
- Existen sub-funciones en internet creadas por usuarios para facilitar tareas en autolisp y estas herramientas deben ser aprovechadas siempre y cuando se de su apropiada apreciación al autor.

9 Bibliografía

Para este trabajo se utilizaron muchas fuentes que no fueron guardadas a lo largo de su desarrollo pues por lo general eran foros explicativos en busca de términos puntuales. Así que se habrá usado la misma página varias veces buscando distinta información.

- Clases del profesor Miguel Ángel Baquero Cortés
- <https://forums.autodesk.com/>. Foros de Autodesk donde la comunidad resuelve diversas dudas puntuales con discusión abierta a la solicitud de alguien.
- <https://documentation.help/AutoLISP-Functions/> Página con algunas ayudas de la documentación de Autolisp que muestra el funcionamiento de diversas funciones.
- http://docs.autodesk.com/ACDMAC/2013/ENU/PDFs/acdmac_2013_autolisp_developers_guide.pdf Guía del desarrollador de Autolisp para AutoCad 2013.
- <http://docs.autodesk.com/ACD/2013/ENU/> Lista de funciones de Autolisp y su forma de uso.
- <https://www.cadtutor.net/> Foro donde algunas personas presentan sus dudas puntuales y la gente de la comunidad ayuda a responderla.
- <http://www.lee-mac.com/> Página creada por un experto de AutoCAD con muchos programas ya elaborados y descargables.

- <https://stackoverflow.com/> Foro de programación (no solo AutoCAD) para resolver dudas puntuales de programación.
- <https://www.afralisp.net/> Página con contenidos para aprender sobre la programación en AutoCAD con ejemplos explicativos en su mayoría.
- Videos de Youtube provisionales que encontrara dependiendo de la duda puntual.