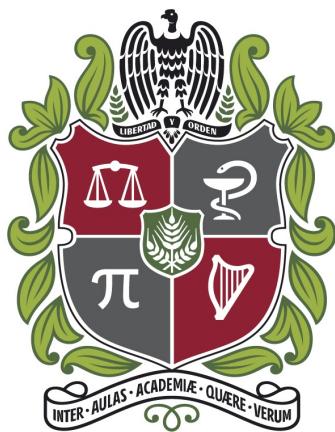


Reloj analogo-digital en Autolips



Juan Sebastián Daleman Martínez
jdaleman@unal.edu.co

Profesor: Miguel Angel Baquero Cortes

Computación gráfica
Departamento de Ingeniería Mecánica y Mecatrónica
Facultad de ingeniería
Universidad Nacional de Colombia - Sede Bogotá
Marzo, 2024

Índice

1. Descripción	3
2. Cálculos matemáticos	3
3. Carga	5
4. Programa	8
5. Resultados	17
6. Conclusiones	17
7. Fuentes de consulta y apoyo	18



1. Descripción

EL programa presentado a continuación permite la ejecución de un reloj de tipo análogo-digital en el cual primero limpia cualquier dibujo creado previamente, luego al usuario le solicita que ingrese:

1. La cantidad de segundos que desea que se ejecute el reloj esto con la finalidad que el programa no se bloquee en un ciclo infinito de funcionamiento y no se presenten problemas por no ser posible parar con la tecla “Escape” la ejecución del programa.
2. Un punto en la ventana del dibujo que se tiene en Autocad o BricsCAD el cual corresponderá con el centro del reloj.
3. Un valor de escala en que se quiere que se dibuje el reloj dado que se tiene como base un reloj de 100 mm como dibujo.

Luego con los datos ingresados por el usuario del programa este procede a escalar el dibujo base a través de una serie de cálculos internos con puntos anteriormente planeados como se muestra en la figura 1. Se genera todos los elementos del reloj, se abstrae del sistema la hora y fecha actuales, se colocan en la posición correcta las manecillas, se colocan los datos de la parte digital, se hace un zoom al dibujo de tal forma que en pantalla se pueda visualizar todo el dibujo correctamente y comienza la animación del reloj hasta que se cumplen los segundos de ejecución que dio el usuario.

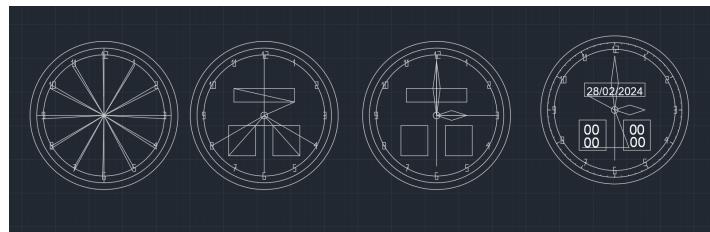


Figura 1: Diagrama base del dibujo del reloj.

2. Cálculos matemáticos

Para el movimiento de las manecillas se calcula cuál debería ser la velocidad angular de cada manecilla en $\left[\frac{\text{rad}}{\text{s}}\right]$ obteniendo los siguientes datos:

- Velocidad segundero: $\frac{2*\pi}{60}$, el cual se obtiene con el comando (setq DelSeg (* 2 (/ pi 60))).
- Velocidad minutero: $\frac{2*\pi}{60*60} = \frac{2*\pi}{3600}$, el cual se obtiene con el comando (setq DelMin (* 2 (/ pi 3600))).
- Velocidad horario: $\frac{2*\pi}{60*60*12} = \frac{2*\pi}{43200}$, el cual se obtiene con el comando (setq DelHor (* 2 (/ pi 43200))).



Con los datos abstractos de la variable de sistema “CDATE” se obtienen la hora actual del sistema luego con estos datos se obtiene la hora actual en su valor en segundos y los minutos actuales en segundos. Para esto se pasan estos valores a segundos, en el caso de los minutos se le suman los segundos y de la hora se suman los segundos y los minutos en segundos. Esto con el fin de tener los datos de tiempo todos en segundos y poder calcular la posición de las manecillas con la velocidad angular de cada una. A continuación se muestran los comandos utilizados

- Minutos en segundos:(setq SegTotalMin (+ Seg (* Minu 60)))
- Hora en segundos: (setq SegTotalHor (+ Seg (* Minu 60) (* Hora 3600)))

para el cálculo de los días de la semana se utilizó la congruencia de Zeller la cual computacionalmente se emplea con la ecuación [1](#)

$$h = \left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + k + \left\lfloor \frac{k}{4} \right\rfloor + \left\lfloor \frac{j}{4} \right\rfloor + 5j \right) \bmod 7 \quad (1)$$

donde se tiene que:

- h es el día de la semana (0 = sábado, 1 = domingo, 2 = lunes, 3 = martes, 4 = miércoles, 5 = jueves, 6 = viernes).
- q es el día del mes.
- m es el mes con la modificación que enero es 13 y febrero es 14 y al hacerse el cálculo con estos dos meses se le resta uno al año actual.
- j es el año dividido en 100.
- k es el año mod 100.

Para la implementación del código no se usaron las modificaciones de año y mes sino que se creó una segunda lista con los datos que correspondiera con el desfase de dos días en donde el cero corresponde al lunes. como se muestra en la figura [2](#).



```
;Funciones para calcular que dia de la semana es apartir de los datos de fecha

(defun c:QueDiaSemanaEs ()
  (setq DiaSemana      (CalcularDiaSemana año mes dia)
        NombreDiaSemana (ObtenerNombreDiaSemana DiaSemana))
)

;Para el calculo de que dia de la semana se esta se usa la conjetura de zeller
;Al ver con pruebas de calculos que la congruencia poseia problemas
;al calcularse con el algoritmo computacional recomendado
;se corrigio generando dos listas para el calculo en vez
;En vez de las midificaciones recomendadas por la ISO en mes y año
(defun CalcularDiaSemana (año mes dia)
  ;(setq mes (if (< mes 3) (+ mes 12) mes)
  ;      año (if (< mes 3) (- año 1) año))
  (setq q dia
        m mes
        K (rem año 100.0)
        J (fix (/ año 100.0))
        h (fix (rem (+ q (fix (/ (* 13 (+ m 1)) 5.0)) K (fix (/ K 4.0))) (fix (/ J 4.0)) (* 5.0 J)) 7))
  )

(defun ObtenerNombreDiaSemana (DiaSemana)
  (if (< mes 3)
    (setq NombresDias '("LU" "MA" "MI" "JU" "VI" "SA" "DO" ))
    (setq NombresDias '("SA" "DO" "LU" "MA" "MI" "JU" "VI" ))
  )
  ;(setq NombresDias ('("Lunes" "Martes" "Miércoles" "Jueves" "Viernes" "Sábado" "Domingo" ))
  (nth DiaSemana NombresDias)
)
```

Figura 2: Código para calculo día de la semana en Autolisp.

3. Carga

Para cargar el programa en Autocad se sigue las siguientes instrucciones:

1. Se abre un nuevo dibujo en Autocad.

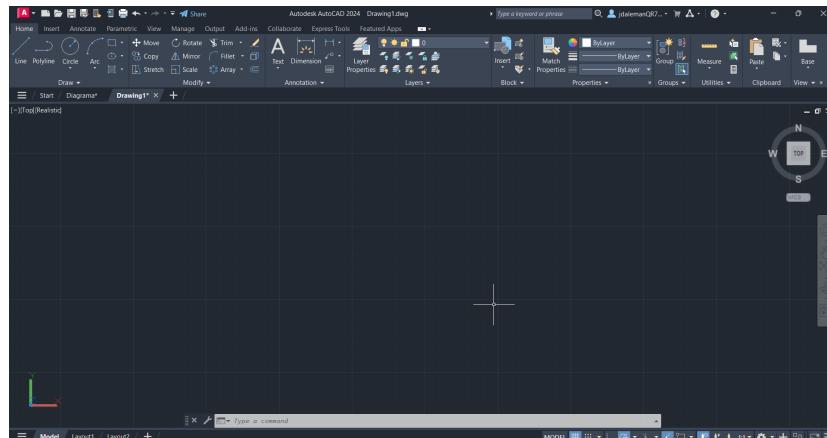


Figura 3: Nuevo dibujo en Autocad.



2. En la consola de Autocad se escribe el comando “Appload” y se da enter.

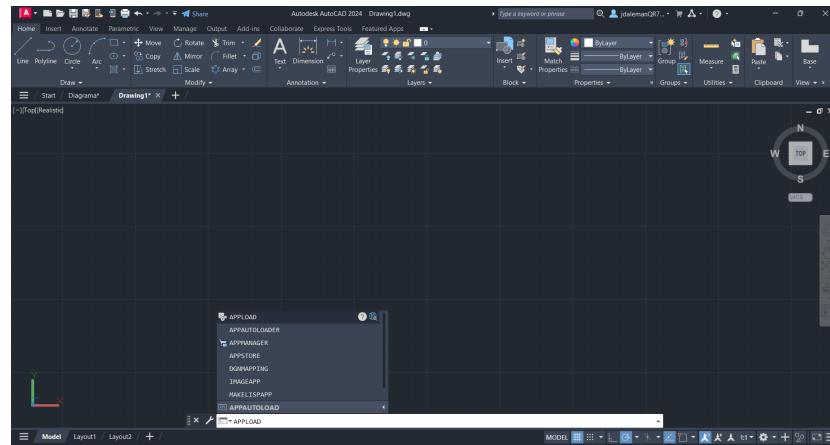


Figura 4: Ingreso del comando “Appload” en la consola de autocad.

3. En la ventana emergente damos click en la carpeta con una flecha verde para navegar en buscador de archivos.

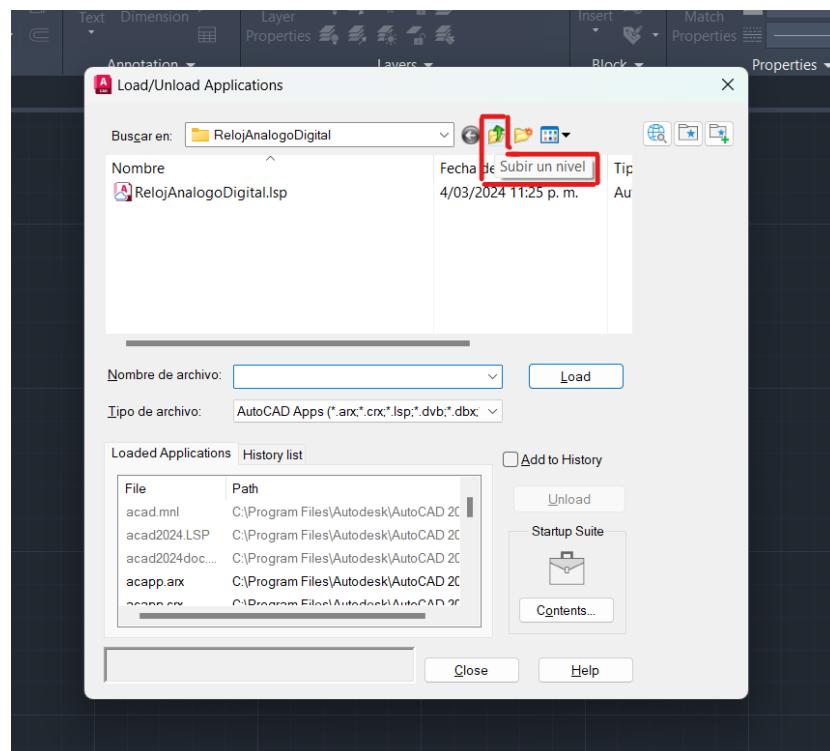


Figura 5: Navegar explorador de archivos desde la ventana emergente de carga.

4. Una vez se ha encontrado donde esta el archivo del programa se selecciona y se da click en load.

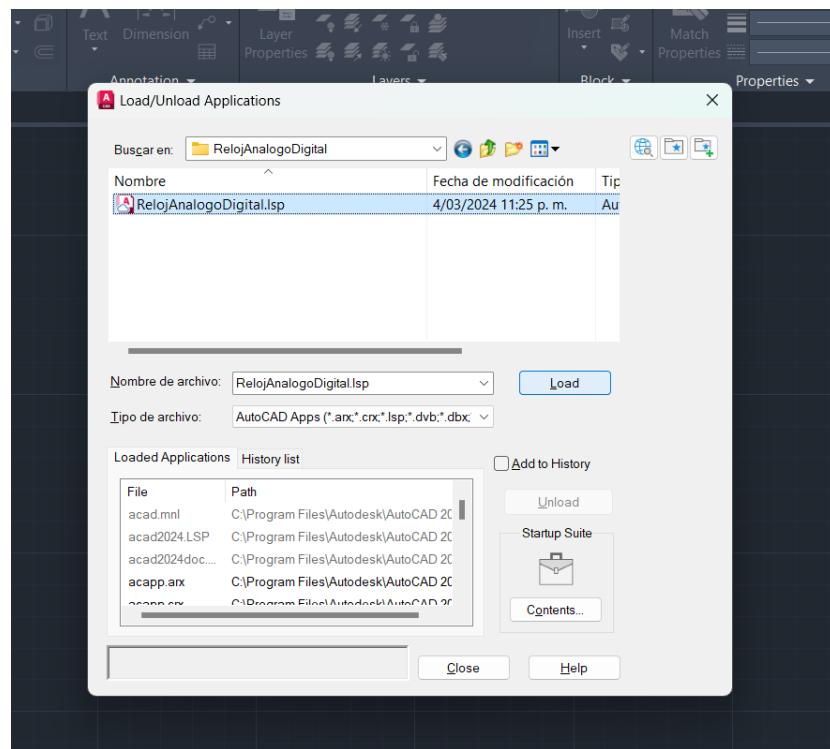


Figura 6: Carga del archivo del programa.

5. por ultimo nos saldrá una ultima ventana emergente en donde podremos decidir si queremos cargar el programa solo para la sesión actual o si para cada vez que se abra Autocad

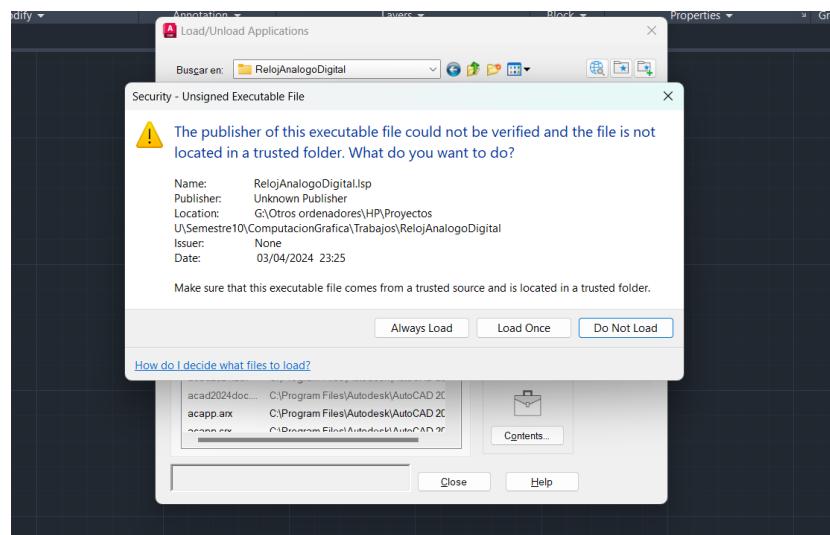


Figura 7: Carga del archivo en Autocad.

6. Ya con esto podemos cerrar la ventana emergente y el programa estara disponible en Autocad.



4. Programa

Para ejecutar el programa del reloj solo tenemos que haber cargado el archivo “RelojAnalogoDigital.lsp” en Autocad, luego llamamos la función Reloj la cual esta como una función de autocad para lo cual solo debemos escribir “Reloj” en la consola de Autocad. Una vez ingresado nos pide primero dar un tiempo el cual esta en segundos, un punto que sera el centro del reloj y una escala que sera un valor de escala para dibujar el reloj (Es importante tener en cuenta que el valor base del radio del reloj es 100 mm por si lo desea más grande o pequeño).



```
1 ;Hacer un reloj analogo-digital que funcione y simule un reloj real.
2 ;Documentado con un doc en word o PDF que tenga el titulo del programa
3 ;con una portada, la descripción, descripción de calculos matematicos,
4 ;una copia del programa documentado, cargado y se corre el programa,
5 ;como queda el reloj despues de correr el programa, conclusiones
6 ;bibliografia de como se hizo el reloj
7 ;se entrega un lsp donde esta el programa
8 ;se envia por correo en un winrar de todos los archivos necesarios
9
10 ;funcion que elimina todos los objetos del dibujo actual
11 (:defun c:borrartodo ()
12   (:command „_ERASE“ „All“ ""))
13 )
14
15 ;funcion que inicializa todo el proceso de hacer el reloj y activarlo
16 (:defun c:Reloj ()
17 )
18
19 ;Se limpia el dibujo de otros elementos no deseados
20 (:c:Borrartodo)
21 ;Se le pide al usuario cual sera
22 ;el tipo de ejecución del reloj
23 ;el centro del reloj
24 ;y la escala teniendo como base un radio de 100
25 ;Se crea una variable de control para el tiempo de ejecución
26   (:setq Parar (:getreal „Cuantos segundos desea que funcione el reloj?\n“)
27     Home (:getpoint „nde el punto central del reloj?\n“)
28     Escala (:getreal „De que escala desea que sea el reloj?\n“)
29     TiempoCorriendo 0
30   )
31
32 ;Se apaga todo el osnap para evitar errores de dibujo
33 (:command „_osnap“ „NONE“)
34 ;Se coloca el custom visual style en realistic
35 (:command „_vscurrent“ „R“)
36
37 ;Se dibujan todos los elementos del reloj analogo
38 (:RelojAnalogo)
39 ;Se dibujan todos los elementos del reloj digital
40 (:RelojDigital)
41
42 ;El bucle se ejecutara infinitamente
43 (:while (not Parar) (loop))
44 ;delay para aumentar el tiempo de 1000 ms
45 (:command „_delay“ 1000)
46 ;Se suma 1 en todas las variables de segundos y la variable de control de tiempo de ejecución
47   (:setq Seg (+ Seg)
48     SegTotalMin (+ SegTotalMin)
49     SegTotalHor (+ SegTotalHor)
50     TextoSegViejo (:assoc 1 DataCambioTextoSeg)
51     TiempoCorriendo (+ TiempoCorriendo 0)
52   )
53
54 ;Se verifica que cuando se cumplen los 60 segundos
55 ;se vuelve a 0 segundos y se suma 1 a minutos
56
57   ((if (= Seg 60)
58     (:setq Seg 0)
59     (:setq Minu (+ Minu Seg) Seg 0))
60   )
61   ;Se verifica que cuando se cumplen los 60 minutos
62   ((if (= Minu 60)
63     (:setq Hora (+ Hora) Minu 0)
64     ;Se verifica que cuando se cumplen las 24 horas
65     ((if (= Hora 24)
66       (:setq Hora 0) ;se vuelve a 0 hora y se le suma 1 al dia
67       (:progn (:setq dia (+ dia) Hora 0)
68         ;Se verifica si se llego a alguna condición de dia para cambio de mes
69         ((if (> dia 27) (:progn
70           (:COND
71             ;Se verifica si el mes es del grupo de 31 dias
72             ((member mes Meses31Dias)
73               ;Si el dia indica que ya paso el 31 se busca actualizar las variables
74               ((if (> dia 31)
75                 (:if (= mes 12)
76                   ;Si el mes es diciembre se hace el cambio de año
77                   ;Se actualizan las variables de la fecha y el año
78                   ;Se actualiza la entidad que muestra el año en el reloj
79                   (:progn (:setq mes 1 dia 1 año (+ año 1))
80                     (ActualizarAñoDig)
81                   )
82                   ;De no ser diciembre se actualizan solo las variables de mes y dia
83                   (:setq mes (+ mes) dia 1)
84                 )
85               )
86             )
87             ;Se verifica si el mes es del grupo de 31 días
88             ((member mes Meses30Dias)
89               ;Si el dia indica que ya paso el 30 se busca actualizar las variables
90               ((if (> dia 30)
91                 (:setq mes (+ mes) dia 1)
92               )
93             )
94             ;Al no estar en los dos grupos anteriores el mes es febrero
95             ;Se verifica con el limite de febrero al ser bisiesto o no si este se revaso
96             ((if (> dia FebreroLim)
97               (:setq mes (+ mes) dia 1)
98             )
99             )
100            ;Se actualiza la entidad que muestra el mes en el reloj
101            (ActualizarMesDig)
102            )
103          )
104        )
105        ;Se actualiza la entidad que muestra el dia y la que muestra que dia de la semana es en el reloj
106        (ActualizarDiaDig)
107        (ActualizarDiaSemanaDig)
108      )
109    )
110    ;Se actualiza la entidad que muestra la hora en el reloj
111    (ActualizarHoraDig)
112  )
113  )
114  ;Se actualiza la entidad que muestra los minutos en el reloj
115  (ActualizarMinDig)
116  )
117  )
118  ;Se actualiza la entidad que muestra los segundos en el reloj
119  (ActualizarSegDig)
120  ;Se actualizan las posiciones de las agujas
121  (:c:ActualizarAgujas)
122  (:command „_regenall“)
123  )
124  )
125
126 ;funcion para abstraer del sistema datos de fecha y hora
127
```

Figura 8: Código Autolips 1.



```
1 ;Funcion para abstraer del sistema datos de fecha y hora
2
3 (defun c:ObtenerFechaYHora ()
4
5 ;Se obtiene la fecha y hora del sistema de la variable CDATE
6 ;de la variable se extraen cada uno de los datos
7 ;se definen tambien las velocidades angulares de cada manecilla en rad/seg
8 ;se calculan las horas y minutos en segundos
9 (setq TimeData (getvar "CDATE")
10 TextTimeData (rtos TimeData 2 10)
11 TextAño (substr TextTimeData 1 4)
12 TextMes (substr TextTimeData 5 2)
13 TextDia (substr TextTimeData 7 2)
14 TextHora (substr TextTimeData 10 2)
15 TextMin (substr TextTimeData 12 2)
16 TextSeg (substr TextTimeData 14 2)
17 año (atoi TextAño)
18 mes (atoi TextMes)
19 dia (atoi TextDia)
20 Hora (atoi TextHora)
21 Minu (atoi TextMin)
22 Seg (atoi TextSeg)
23 DelSeg (* 2 (/ pi 60))
24 DelMin (* 2 (/ pi 3600))
25 DelHor (* 2 (/ pi 43200))
26 SegTotalMin (+ Seg (* Minu 60))
27 SegTotalHor (+ Seg (* Minu 60) (* Hora 3600))
28 Meses31Dias '(1 3 5 7 8 10 12)
29 Meses30Dias '(4 6 9 11)
30 )
31 ;Se verifica si el año es bisiesto o no
32 ;y se pone el límite de los días de febrero según
33 ;sea el caso
34 (if (= (rem año 4) 0)
35 (setq FebreroLim 29)
36 (setq FebreroLim 28)
37 )
38 )
39 )
40
41
42 ;Funciones para calcular que dia de la semana es apartir de los datos de fecha
43
44 (defun c:QueDiaSemanaEs ()
45 (setq DiaSemana (CalcularDiaSemana año mes dia)
46 NombreDiaSemana (ObtenerNombreDiaSemana DiaSemana)
47 )
48 )
49 )
50
51 ;Para el calculo de que dia de la semana se esta se usa la conjetura de zeller
52 ;Al ver con pruebas de calculos que la congruencia poseia problemas
53 ;al calcularse con el algoritmo computacional recomendado
54 ;se corrigio generando dos listas para el calculo en vez
55 ;En vez de las modificaciones recomendadas por la ISO en mes y año
56 (defun CalcularDiaSemana (año mes dia)
57 ;(setq mes (if (< mes 3) (+ mes 12) mes)
58 ;      año (if (< mes 3) (- año 1) año))
59 (setq q dia
60       m mes
61       K (rem año 100.0)
62       J (fix (/ año 100.0))
63       h (fix (rem (+ q (fix (/ (* 13 (+ m 1)) 5.0)) K (fix (/ K 4.0))) (fix (/ J 4.0)) (* 5.0 J)) 7))
64 )
65 )
66
67 (defun ObtenerNombreDiaSemana (DiaSemana)
68 (if (< mes 3)
69 (setq NombresDias ("LU" "MA" "MI" "JU" "VI" "SA" "DO" ))
70 (setq NombresDias ('("SA" "DO" "LU" "MA" "MI" "JU" "VI" ))
71 )
72 ;(setq NombresDias ('("Lunes" "Martes" "Miércoles" "Jueves" "Viernes" "Sábado" "Domingo" ))
73 (nth DiaSemana NombresDias)
74 )
```

Figura 9: Código Autolips 2.



Computación gráfica 2024-1

```
1 ;Dibuja la parte digital del reloj y pone los datos correspondientes
2 (defun RelojDigital ()
3   (DibCajas)
4   (PonerDatosDigitales)
5 )
6
7 ;Dibuja las cajas donde estan los números del reloj digital
8 ;donde se indicara que dia de la semana es
9 ;y la fecha del dia
10 (defun DibCajas (/ p1a p2a p1b p2b p1c p2c)
11   (setq p1a (polar Home (* -131 (/ pi 180)) (* Escala 20))
12     p2a (polar p1a (* -132 (/ pi 180)) (* Escala 60))
13     p1b (polar Home (* -49 (/ pi 180)) (* Escala 20))
14     p2b (polar p1b (* -48 (/ pi 180)) (* Escala 60))
15     p1c (polar Home (* 24 (/ pi 180)) (* Escala 49))
16     p2c (polar p1c (* 167 (/ pi 180)) (* Escala 92)))
17   )
18
19   (command "_.rectang" p1a p2a "" "_.rectang" p1b p2b "" "_.rectang" p1c p2c ""))
20 )
21
22 ;Escribe los textos para el reloj digital
23 (defun PonerDatosDigitales (/ pSeg pMin pHora pDia pFecha)
24   (setq pSeg (polar Home (* -70 (/ pi 180)) (* Escala 60.5))
25     pMin (polar Home (* -125 (/ pi 180)) (* Escala 72))
26     pHora (polar pMin (* 90 (/ pi 180)) (* Escala 20))
27     pDiaSemana (polar pSeg (* 101 (/ pi 180)) (* Escala 28.5))
28     pDia (polar Home (* 153 (/ pi 180)) (* Escala 48.8))
29     pRay1 (polar pDia (* 0 (/ pi 180)) (* Escala 20))
30     pMes (polar pRay1 (* 0 (/ pi 180)) (* Escala 5))
31     pRay2 (polar pMes (* 0 (/ pi 180)) (* Escala 20))
32     pAño (polar pRay2 (* 0 (/ pi 180)) (* Escala 5))
33     Altura1 (* Escala 15)
34     Altura2 (* Escala 12)
35     Rotacion 0.0
36   )
37
38 ;Se crea el cuadro de texto para segundos
39 (if (< (strlen TextSeg) 2)
40   (setq TextSeg (strcat "0" TextSeg))
41 )
42 (Command ".text" pSeg Altura1 Rotacion TextSeg "" "")
43 (setq CuadroTextoSeg (entlast)
44       DatosCuadroTextoSeg (entget CuadroTextoSeg)
45 )
46
47 ;Se crea el cuadro de texto para minutos
48 (Command ".text" pMin Altura1 Rotacion TextMin "" "")
49 (setq CuadroTextoMin (entlast)
50       DatosCuadroTextoMin (entget CuadroTextoMin)
51 )
52
53 ;Se crea el cuadro de texto para hora
54 (Command ".text" pHora Altura1 Rotacion TextHora "" "")
55 (setq CuadroTextoHora (entlast)
56       DatosCuadroTextoHora (entget CuadroTextoHora)
57 )
58
59 ;Se crea el cuadro de texto para dia de la semana
60 (Command ".text" pDiaSemana Altura1 Rotacion NombreDiaSemana "" "")
61 (setq CuadroTextoDiaSemana (entlast)
62       DatosCuadroTextoDiaSemana (entget CuadroTextoDiaSemana)
63 )
64
65 ;Se crea el cuadro de texto para dia
66 (Command ".text" pDia Altura2 Rotacion TextDia "" "")
67 (setq CuadroTextoDia (entlast)
68       DatosCuadroTextoDia (entget CuadroTextoDia)
69 )
70
71 ;Se crea el cuadro de texto para la / que divide dia y mes
72 (Command ".text" pRay1 Altura2 Rotacion "/" "" "")
73
74 ;Se crea el cuadro de texto para mes
75 (Command ".text" pMes Altura2 Rotacion TextMes "" "")
76 (setq CuadroTextoMes (entlast)
77       DatosCuadroTextoMes (entget CuadroTextoMes)
78 )
79
80 ;Se crea el cuadro de texto para la / que divide mes y año
81 (Command ".text" pRay2 Altura2 Rotacion "/" "" "")
82
83 ;Se crea el cuadro de texto para año
84 (Command ".text" pAño Altura2 Rotacion TextAño "" "")
85 (setq CuadroTextoAño (entlast)
86       DatosCuadroTextoAño (entget CuadroTextoAño))
87
88
89 )
```

Figura 10: Código Autolips 3.



```
1 ;Funcion que actualiza los segundos en el reloj digital
2 (defun ActualizarSegDig ()
3   (setq TextSeg (itoa Seg))
4   (if (< (strlen TextSeg) 2)
5     (setq TextSeg (strcat "0" TextSeg))
6   )
7   (setq TextSegNuevo (cons 1 TextSeg)
8         TextoSegViejo (assoc 1 DatosCuadroTextoSeg)
9         DatosCuadroTextoSegNuevo (subst TextSegNuevo TextoSegViejo DatosCuadroTextoSeg)
10    )
11   (entmod DatosCuadroTextoSegNuevo)
12  )
13
14 ;Funcion que actualiza los minutos en el reloj digital
15 (defun ActualizarMinDig ()
16   (setq TextMin (itoa Minu))
17   (if (< (strlen TextMin) 2)
18     (seta TextMin (strcat "0" TextMin))
19   )
20   (setq TextMinNuevo (cons 1 TextMin)
21         TextoMinViejo (assoc 1 DatosCuadroTextoMin)
22         DatosCuadroTextoMinNuevo (subst TextMinNuevo TextoMinViejo DatosCuadroTextoMin)
23    )
24   (entmod DatosCuadroTextoMinNuevo)
25  )
26
27 ;Funcion que actualiza la hora en el reloj digital
28 (defun ActualizarHoraDig ()
29   (setq TextHora (itoa Hora))
30   (if (< (strlen TextHora) 2)
31     (seta TextHora (strcat "0" TextHora))
32   )
33   (setq TextHoraNuevo (cons 1 TextHora)
34         TextoHoraViejo (assoc 1 DatosCuadroTextoHora)
35         DatosCuadroTextoHoraNuevo (subst TextHoraNuevo TextoHoraViejo DatosCuadroTextoHora)
36    )
37   (entmod DatosCuadroTextoHoraNuevo)
38  )
39
40 ;Funcion que actualiza el dia en el reloj digital
41 (defun ActualizarDiaDig ()
42   (setq TextDia (itoa dia))
43   (if (< (strlen TextDia) 2)
44     (setq TextDia (strcat "0" TextDia))
45   )
46   (setq TextDiaNuevo (cons 1 TextDia)
47         TextoDiaViejo (assoc 1 DatosCuadroTextoDia)
48         DatosCuadroTextoDiaNuevo (subst TextDiaNuevo TextoDiaViejo DatosCuadroTextoDia)
49    )
50   (entmod DatosCuadroTextoDiaNuevo)
51  )
52
53 ;Funcion que actualiza el dia de la semana en el reloj digital
54 (defun ActualizarDiaSemanaDig ()
55   (c:QueDiaSemanaEs)
56   (setq TextDiaSemanaNuevo (cons 1 NombreDiaSemana)
57         TextoDiaSemanaViejo (assoc 1 DatosCuadroTextoDiaSemana)
58         DatosCuadroTextoDiaSemanaNuevo (subst TextDiaSemanaNuevo TextoDiaSemanaViejo DatosCuadroTextoDiaSemana)
59    )
60   (entmod DatosCuadroTextoDiaSemanaNuevo)
61  )
62
63 ;Funcion que actualiza el mes en el reloj digital
64 (defun ActualizarMesDig ()
65   (setq TextMes (itoa mes))
66   (if (< (strlen TextMes) 2)
67     (seta TextMes (strcat "0" TextMes))
68   )
69   (setq TextMesNuevo (cons 1 TextMes)
70         TextoMesViejo (assoc 1 DatosCuadroTextoMes)
71         DatosCuadroTextoMesNuevo (subst TextMesNuevo TextoMesViejo DatosCuadroTextoMes)
72    )
73   (entmod DatosCuadroTextoMesNuevo)
74  )
75
76 ;Funcion que actualiza el año en el reloj digital
77 (defun ActualizarAñoDig ()
78   (setq TextAño (itoa año))
79   (setq TextAñoNuevo (cons 1 TextAño)
80         TextoAñoViejo (assoc 1 DatosCuadroTextoAño)
81         DatosCuadroTextoAñoNuevo (subst TextAñoNuevo TextoAñoViejo DatosCuadroTextoAño)
82    )
83   (entmod DatosCuadroTextoAñoNuevo)
84  )
```

Figura 11: Código Autolips 4.



```
1 ;Dibuja los elementos del reloj analogo
2 (defun RelojAnalogo ( / p1a p2a p3a)
3
4   (setq p1a (polar Home (* 153 (/ pi 180)) (* Escala 88.9))
5         p2a (polar Home (* 123 (/ pi 180)) (* Escala 86.5))
6         p3a (polar Home (* 91 (/ pi 180)) (* Escala 85)))
7
8   ;Dibuja el marco del reloj
9   (DibujarReloj)
10  ;Dibuja los numeros del 1-9
11  (c:D1b1 Home 58 85)
12  (c:D1b2 Home 26 90)
13  (c:D1b3 Home -3 92)
14  (c:D1b4 Home -33 93)
15  (c:D1b5 Home -63 93.5)
16  (c:D1b6 Home -91 95)
17  (c:D1b7 Home -118 94)
18  (c:D1b8 Home -148 94.3)
19  (c:D1b9 Home -177 92)
20  ;Dibuja el 10
21  (c:D1b1 Home 153 88.9)
22  (c:D1b2 p1a 0 3)
23  ;Dibuja el 11
24  (c:D1b1 Home 123 86.5)
25  (c:D1b1 p2a 0 4)
26  ;Dibuja el 12
27  (c:D1b1 Home 91 85)
28  (c:D1b2 p3a 0 7)
29  ;Dibuja las agujas del reloj
30  (DibAgujas)
31  ;Se hace zoom al reloj para que el usuario lo visualice correctamente
32  ;Se regenera los elementos creados para evitar errores
33  (command "zoom" "extents")
34  (command "regen")
35  ;Se obtiene la hora y fecha del sistema
36  ;,se calcula el dia de la semana y
37  ;se actualizan las posiciones de las agujas
38  (c:ObtenerFechaYHora)
39  (c:QuediaSemanaEs)
40  (c:ActualizarAgujas)
41 )
42
43 ;Dibuja el marco del reloj y el circulo del centro con sus decoraciones
44 (defun DibujarReloj ()
45   (setq radio (* Escala 100)
46         p1dec (polar Home (* 98 (/ pi 180)) (* Escala 100))
47         p2dec (polar Home (* 98 (/ pi 180)) (* Escala 95))
48         p3dec (polar Home (* 98 (/ pi 180)) (* Escala 97.5))
49   )
50
51   (command ".circle" Home (+ radio (* Escala 10)) "" ".circle" Home radio "" ".circle" Home (* Escala 5) "")
52   (command "_line" p1dec p2dec "")
53   (setq Lin1Dec (entlast))
54   (setq DatosLin1Dec (entget Lin1Dec ))
55   (command ".ARAYPOLAR" Lin1Dec "" Home 12 360 "X")
56   (command ".line" p1dec p3dec "")
57   (setq Lin2Dec (entlast))
58   (setq DatosLin2Dec (entget Lin2Dec ))
59   (command ".ARAYPOLAR" Lin2Dec "" Home 60 360 "X")
60 )
```

Figura 12: Código Autolips 5.



```
1 ;Dibuja todas las manecillas mirando a las 12
2 (defun DibAgujas ()
3   ;Define los puntos iniciales de cada uno de los puntos de las manecillas
4   (setq p1Seg (polar Home (* 90 (/ pi 180)) (* Escala 75))
5         p1Min (polar Home (* 98 (/ pi 180)) (* Escala 40.5))
6         p2Min (polar p1Min (* 82 (/ pi 180)) (* Escala 40))
7         p3Min (polar p2Min (* -82 (/ pi 180)) (* Escala 40))
8         p1Hora (polar Home (* 108 (/ pi 180)) (* Escala 23.5))
9         p2Hora (polar p1Hora (* 72 (/ pi 180)) (* Escala 23.7))
10        p3Hora (polar p2Hora (* -72 (/ pi 180)) (* Escala 23.7)))
11   ;Define la distancia desde el centro a cada uno de los puntos
12   r1Seg (distance Home p1Seg)
13   r1Min (distance Home p1Min)
14   r2Min (distance Home p2Min)
15   r3Min (distance Home p3Min)
16   r1Hora (distance Home p1Hora)
17   r2Hora (distance Home p2Hora)
18   r3Hora (distance Home p3Hora)
19   ;Define el angulo de la linea que se forma del centro y cada uno de los puntos
20   Ang1Seg (angle Home p1Seg)
21   Ang1Min (angle Home p1Min)
22   Ang2Min (angle Home p2Min)
23   Ang3Min (angle Home p3Min)
24   Ang1Hora (angle Home p1Hora)
25   Ang2Hora (angle Home p2Hora)
26   Ang3Hora (angle Home p3Hora)
27 )
28
29 ;Dibuja el segundero
30 (command "_line" Home p1Seg "")
31 (setq Segundero (entlast)
32       DatosSegundero (entget Segundero)
33       Mov1Seg (assoc 11 DatosSegundero)
34 )
35 )
36 (setq DatosSegundero (CambiarColorEntidad DatosSegundero 1))
37 (entmod DatosSegundero)
38
39 ;Dibuja el minutero
40 (command "_pline" Home p1Min p2Min p3Min Home "")
41 (setq Minutero (entlast)
42       DatosMinutero (entget Minutero)
43       PuntosMinutero (ObtenerPuntosPolilinea DatosMinutero)
44 )
45 )
46 (setq DatosMinutero (CambiarColorEntidad DatosMinutero 2))
47 (entmod DatosMinutero)
48
49 ;Dibuja el horario
50 (command "_pline" Home p1Hora p2Hora p3Hora Home "")
51 (setq Horario (entlast)
52       DatosHorario (entget Horario)
53       PuntosHorario (ObtenerPuntosPolilinea DatosHorario)
54 )
55 (setq DatosHorario (CambiarColorEntidad DatosHorario 120))
56 (entmod DatosHorario)
57
58 )
```

Figura 13: Código Autolips 6.



```
● ● ●

1 ;Actualiza la posicion de los puntos que describen las manecillas
2 ;del reloj con respecto a la hora actual que se tiene de CDATE
3
4 (defun c:ActualizarAgujas ()
5   ;Encuentra cual debe ser el angulo que debe tener cada manecilla
6   (setq AngSeg (* DelSeg Seg)
7     AngMin (* DelMin SegTotalMin)
8     AngHora (* DelHor SegTotalHor)
9
10    Elemfalt (list 10)
11    Elemfalt1 (list 11)
12    ModHome (Append Elemfalt Home)
13    ;Se Encuentra las nuevas posiciones de cada punto
14    ;y se crean las listas que se remplazarán en la entidad
15    piSeg (polar Home (- Ang1Seg AngSeg) r1Seg)
16    piSeg (Append Elemfalt1 piSeg)
17
18    p1Min (polar Home (- Ang1Min AngMin) r1Min)
19    p2Min (polar Home (- Ang2Min AngMin) r2Min)
20    p3Min (polar Home (- Ang3Min AngMin) r3Min)
21    p1Min (Append Elemfalt p1Min)
22    p2Min (Append Elemfalt p2Min)
23    p3Min (Append Elemfalt p3Min)
24    NewPuntosMinutero (list ModHome p1Min p2Min p3Min ModHome)
25
26    piHora (polar Home (- Ang1Hora AngHora) r1Hora)
27    p2Hora (polar Home (- Ang2Hora AngHora) r2Hora)
28    p3Hora (polar Home (- Ang3Hora AngHora) r3Hora)
29    piHora (Append Elemfalt piHora)
30    p2Hora (Append Elemfalt p2Hora)
31    p3Hora (Append Elemfalt p3Hora)
32    NewPuntosHorario (list ModHome piHora p2Hora p3Hora ModHome)
33
34    ;Sustituyen los valores de cada entidad
35    DatosSegundero2 (subst piSeg Mov1Seg DatosSegundero)
36    DatosMinutero2 (ReemplazarPuntos PuntosMinutero NewPuntosMinutero DatosMinutero)
37    DatosHorario2 (ReemplazarPuntos PuntosHorario NewPuntosHorario DatosHorario)
38  )
39
40  ;Se actualiza como se presenta en el dibujo las manecillas
41  (entmod DatosSegundero2)
42  (entmod DatosMinutero2)
43  (entmod DatosHorario2)
44
45  )
46
47
48 (defun ObtenerPuntosPolilinea (plineData)
49
50   (setq puntos '()) ; Inicializar una lista para almacenar los puntos
51
52   ; Iterar sobre las propiedades para encontrar los puntos
53   (foreach prop plineData
54     (if (= (car prop) 10) ; Verificar si el código de grupo es 10 (representa un punto)
55       (setq puntos (cons prop puntos)) ; Agregar las coordenadas del punto a la lista
56     )
57   )
58
59   ;Retorno de los puntos de la polilínea
60   puntos
61 )
62
63
64 ;Reemplaza los punto de una polilínea por los nuevos puntos en la entidad ingresada
65 (defun ReemplazarPuntos (Puntos NewPuntos DatosEntidad)
66   (setq NewDatosEntidad DatosEntidad)
67   (foreach Punto Puntos
68     (setq NewPunto (car NewPuntos)
69       NewDatosEntidad (subst NewPunto Punto NewDatosEntidad)
70       NewPuntos (cdr NewPuntos)
71     )
72   )
73   ; Retornamos el resultado final
74   NewDatosEntidad
75 )
76
77 ;Cambia los datos de color de una entidad por un color definido
78 (defun CambiarColorEntidad (DatosEntidad Color)
79   (setq NewColor (list (cons 62 Color)))
80   (setq OldColor (assoc 62 DatosEntidad))
81   (if OldColor
82     (setq NewDatosLinea (subst NewColor OldColor DatosEntidad))
83     (setq NewDatosLinea (append DatosEntidad NewColor))
84   )
85   NewDatosLinea
86 )
```

Figura 14: Código Autolips 7.



Computación gráfica 2024-1

```
1 ;Funciones para dibujar todos los dígitos de 0-9 dando un punto inicial y un offset polar a ese punto
2
3 (defun c:Dib0 (Home Ang Dis / p1a p2a p3a pda)
4   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
5   p2a (polar p1a (* 0 (/ pi 180)) (* Escala 4))
6   p3a (polar p1a (* 90 (/ pi 180)) (* Escala 10))
7   p4a (polar p1a (* 90 (/ pi 180)) (* Escala 10))
8 )
9
10  (command "_pline" p1a p2a p3a p1a "")
11 )
12
13 (defun c:Dib1 (Home Ang Dis / p1a p2a p3a)
14   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
15   p2a (polar p1a (* 90 (/ pi 180)) (* Escala 10))
16   p3a (polar p2a (* 240 (/ pi 180)) (* Escala 4))
17 )
18
19  (command "_pline" p1a p2a p3a "")
20 )
21
22 (defun c:Dib2 (Home Ang Dis / p1a p2a p3a p3a p6a)
23   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
24   p2a (polar p1a (* 180 (/ pi 180)) (* Escala 4))
25   p3a (polar p2a (* 90 (/ pi 180)) (* Escala 5))
26   p4a (polar p3a (* 0 (/ pi 180)) (* Escala 4))
27   p5a (polar p4a (* 90 (/ pi 180)) (* Escala 5))
28   p6a (polar p5a (* 180 (/ pi 180)) (* Escala 4))
29 )
30
31
32  (command "_pline" p1a p2a p3a p4a p5a p6a "")
33 )
34
35 (defun c:Dib3 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
36   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
37   p2a (polar p1a (* 180 (/ pi 180)) (* Escala 4))
38   p3a (polar p2a (* 90 (/ pi 180)) (* Escala 5))
39   p4a (polar p3a (* 0 (/ pi 180)) (* Escala 4))
40   p5a (polar p4a (* 90 (/ pi 180)) (* Escala 5))
41   p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
42 )
43
44
45  (command "_pline" p5a p6a p4a p3a p4a p1a p2a "")
46 )
47
48 (defun c:Dib4 (Home Ang Dis / p1a p2a p3a p4a)
49   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
50   p2a (polar p1a (* 90 (/ pi 180)) (* Escala 10))
51   p3a (polar p2a (* -120 (/ pi 180)) (* Escala 6.4))
52   p4a (polar p3a (* 0 (/ pi 180)) (* Escala 5.5))
53 )
54
55
56  (command "_pline" p1a p2a p3a p4a "")
57 )
58
59 (defun c:Dib5 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
60   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
61   p2a (polar p1a (* 0 (/ pi 180)) (* Escala 4))
62   p3a (polar p2a (* 90 (/ pi 180)) (* Escala 5))
63   p4a (polar p3a (* 180 (/ pi 180)) (* Escala 4))
64   p5a (polar p4a (* 90 (/ pi 180)) (* Escala 5))
65   p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
66 )
67
68
69  (command "_pline" p1a p2a p3a p4a p5a p6a "")
70 )
71
72 (defun c:Dib6 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
73   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
74   p2a (polar p1a (* 180 (/ pi 180)) (* Escala 4))
75   p3a (polar p2a (* 90 (/ pi 180)) (* Escala 5))
76   p4a (polar p3a (* 0 (/ pi 180)) (* Escala 4))
77   p5a (polar p4a (* 90 (/ pi 180)) (* Escala 5))
78   p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
79 )
80
81
82  (command "_pline" p6a p5a p1a p2a p3a p4a "")
83 )
84
85 (defun c:Dib7 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
86   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
87   p2a (polar p1a (* 70 (/ pi 180)) (* Escala 10.5))
88   p3a (polar p2a (* 180 (/ pi 180)) (* Escala 4))
89   p4a (polar p3a (* 70 (/ pi 180)) (* Escala 5.25))
90   p5a (polar p4a (* -90 (/ pi 180)) (* Escala 4.95))
91   p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
92 )
93
94
95  (command "_pline" p1a p2a p3a p2a p4a p5a p6a "")
96 )
97
98 (defun c:Dib8 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
99   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
100  p2a (polar p1a (* 0 (/ pi 180)) (* Escala 4))
101  p3a (polar p1a (* 90 (/ pi 180)) (* Escala 5))
102  p4a (polar p3a (* 0 (/ pi 180)) (* Escala 4))
103  p5a (polar p3a (* 90 (/ pi 180)) (* Escala 5))
104  p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
105 )
106
107
108  (command "_pline" p5a p1a p2a p6a p5a p3a p4a "")
109 )
110
111 (defun c:Dib9 (Home Ang Dis / p1a p2a p3a p4a p5a p6a)
112   (setq p1a (polar Home (* Ang (/ pi 180)) (* Escala Dis)))
113   p2a (polar p1a (* 0 (/ pi 180)) (* Escala 4))
114   p3a (polar p1a (* 90 (/ pi 180)) (* Escala 5))
115   p4a (polar p3a (* 0 (/ pi 180)) (* Escala 4))
116   p5a (polar p3a (* 90 (/ pi 180)) (* Escala 5))
117   p6a (polar p5a (* 0 (/ pi 180)) (* Escala 4))
118 )
119
119  (command "_pline" p1a p2a p6a p5a p3a p4a "")
120 )
121
```

Figura 15: Código Autolips 8.



5. Resultados

Como se puede observar en la figura 16 se obtuvo un reloj no solo estéticamente que se ve bien sino que también posee diferentes funcionalidades con fecha y hora. Así como diferentes colores en cada una de las manecillas del reloj.

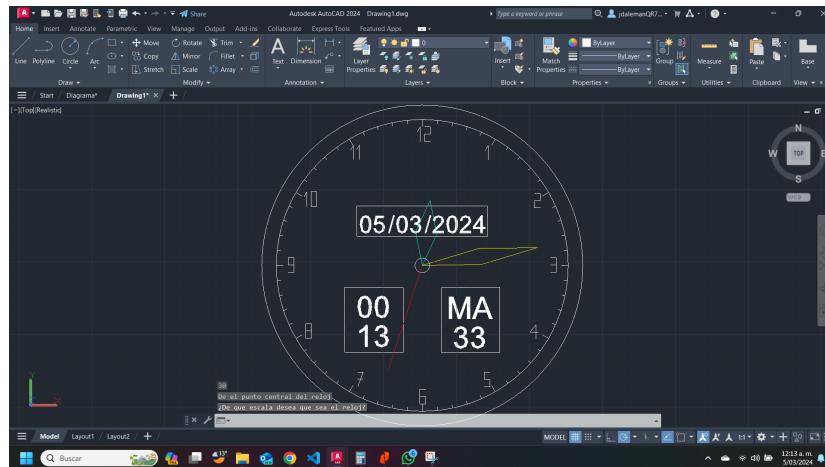


Figura 16: Reloj en dibujado y resultado.

6. Conclusiones

- Autolips es un lenguaje muy versátil para generar parametrizaciones de dibujos, lo cual permite el manejo fácil de tareas repetitivas en dibujos que pueden solucionarse con programación.
- Procesos que en Autocad usando herramientas donde la persona tiene que intervenir todo el tiempo pueden ser mejorados con el lenguaje de Autolips para generar mayor rápidas en el proceso o incluso tareas que no serían reproducibles por una persona como animaciones.
- El manejo de diferentes comandos de Autolips facilita cada vez más la tarea del programador simplificando cierto manejo de datos y parámetros, permitiendo hacer tareas con autocad muy entretenidas.
- La aplicación de programación de autolips con un reloj permite conocer mucho el manejo de entidades y herramientas de programación básicas como operadores lógicos y numéricos, ciclos y herramientas propias del lenguaje de Autolips como las entidades y manejo de datos en DXF.

Índice de figuras

1. Diagrama base del dibujo del reloj.	3
2. Código para calculo día de la semana en Autolisp.	5



3.	Nuevo dibujo en Autocad.	5
4.	Ingreso del comando “Upload” en la consola de autocad.	6
5.	Navegar explorador de archivos desde la ventana emergente de carga.	6
6.	Carga del archivo del programa.	7
7.	Carga del archivo en Autocad.	7
8.	Código Autolips 1.	9
9.	Código Autolips 2.	10
10.	Código Autolips 3.	11
11.	Código Autolips 4.	12
12.	Código Autolips 5.	13
13.	Código Autolips 6.	14
14.	Código Autolips 7.	15
15.	Código Autolips 8.	16
16.	Reloj en dibujado y resultado.	17

7. Fuentes de consulta y apoyo

- Cardenas, Luis. Curso de Udemy PROGRAMAR EN AUTOCAD CON .^AUTOLISP- RECURSOS DESCARGABLES.
<https://www.udemy.com/course/autolisp-recursos-descargables-para-autocad/>
- ChaGPT. Apoyo en comprencción de la congruencia de zeller para obtención del dia de la semana y calculos hechos en esta congruencia.