

# *AutoLISP* – Programas Exemplo

João Manuel R. S. Tavares  
Joaquim Oliveira Fonseca



Universidade  
do Porto

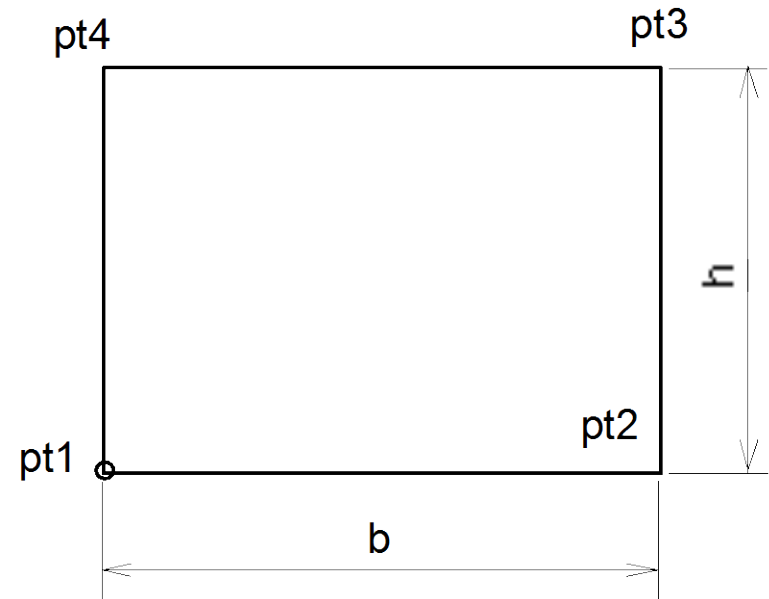
Faculdade de  
Engenharia

**FEUP**

DEMEC  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

# 1º Exemplo - Rectângulo

Desenvolva um programa em AutoLISP que permita criar a figura junta. Utilize a função LIST para definir as coordenadas dos novos pontos.



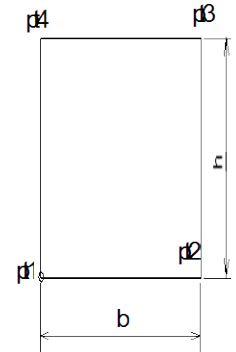
*(O utilizador define a largura, a altura e o ponto pt1.)*

# 1º Exemplo - Rectângulo

```
; Programa para desenhar um rectângulo através de:  
; canto inferior esquerdo, largura e altura  
;
```

```
(Defun c:rect()  
  ; LEITURA  
  ; vai ler os valores de definição  
  (setq b (getreal "\nLargura : "))  
  (setq h (getreal "\nAltura : "))  
  (setq pt1 (getpoint "\nCanto Inferior Esquerdo : "))  
  ; CÁLCULOS  
  ; vai calcular os outros pontos necessários  
  (setq pt2 (list (+ (car pt1) b) (cadr pt1)))  
  (setq pt3 (list (car pt2) (+ (cadr pt1) h)))  
  (setq pt4 (list (car pt1) (cadr pt3)))  
  ; DESENHO  
  ; Vai desenhar através do comando LINE e opção final Close  
  (command "line" pt1 pt2 pt3 pt4 "c")  
)
```

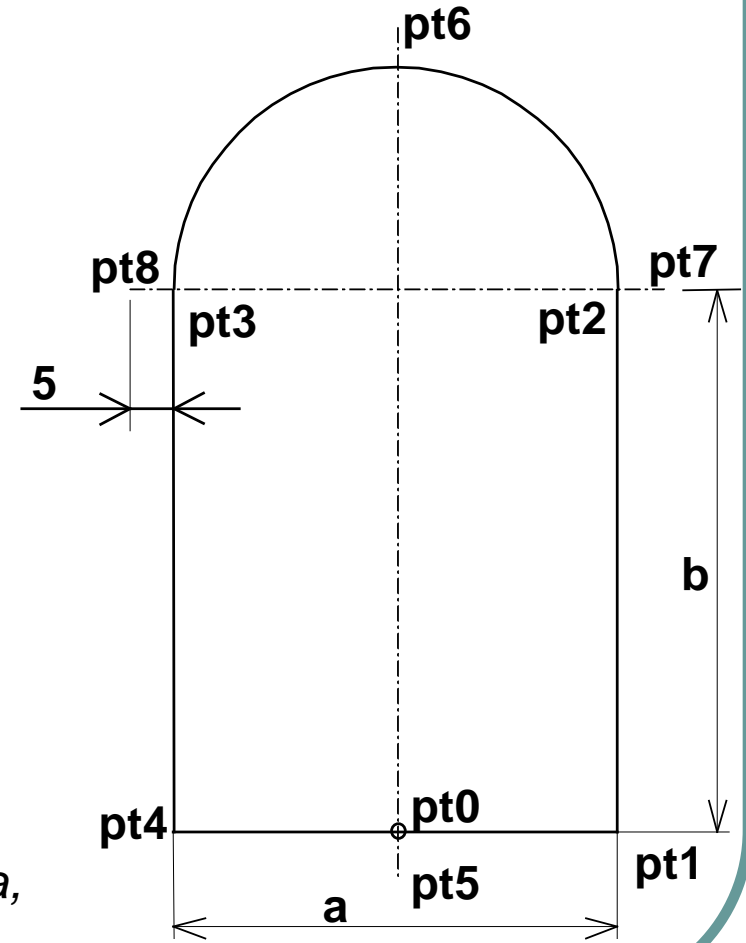
Desenhe um programa em AutoLISP que  
permaneça a figura junta. Utilize a função  
LIST para definir as coordenadas dos  
pontos



# 2º Exemplo - Porta

Desenvolva um programa em AutoLISP que permita criar a figura junta e traçar as linhas a traço misto. Considere que as linhas a traço misto ultrapassam a figura em 5mm.

*(O utilizador define a largura, a altura e o ponto pt0.)*



# 2º Exemplo - Porta

; Programa para desenhar uma Porta de arco através de:

; Centro inferior, largura e altura direita. jof-92-11-25

(Defun c:porta())

; LEITURA

; vai ler os valores

(setq a (getreal "\nLargura da porta : "))

(setq b (getreal "\nAltura direita da porta : "))

(setq pt0 (getpoint "\nPonto Central inferior : "))

; CÁLCULOS

; vai calcular outros pontos necessários ao desenho da figura

(setq a2 (/ a 2))

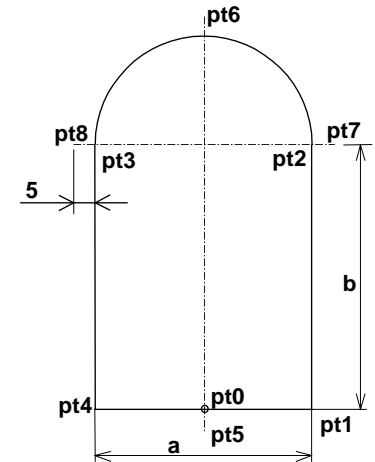
(setq pi2 (/ pi 2))

(setq pt1 (polar pt0 0 a2))

(setq pt2 (polar pt1 pi2 b))

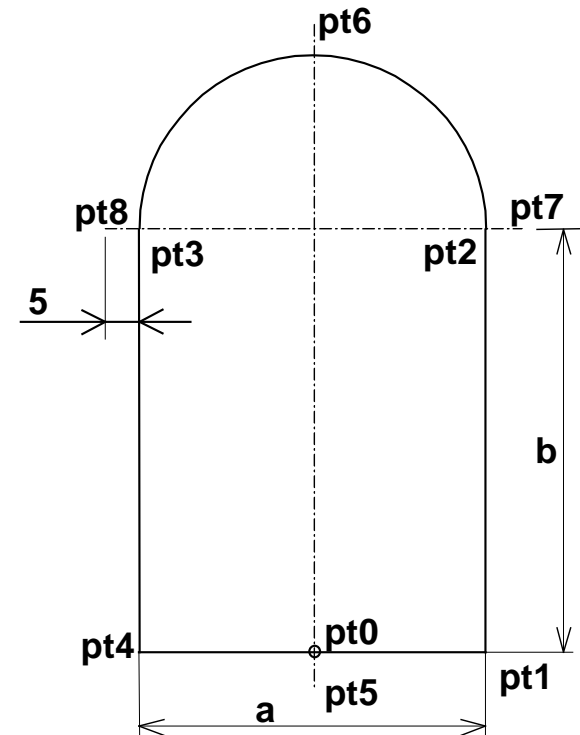
(setq pt4 (polar pt0 pi a2))

(setq pt3 (polar pt4 pi2 b))



# 2º Exemplo - Porta

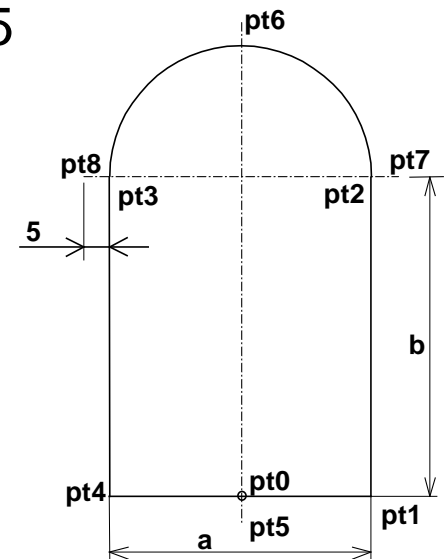
```
(setq pt5 (polar pt0 (- pi2) 5))
(setq pt7 (polar pt2 0 5))
(setq pt8 (polar pt3 pi 5))
(setq pp (+ b a2 10))
(setq pt6 (polar pt5 pi2 pp))
; DESENHO
; Altera o layer e desenha
(command "layer" "m" "TG" "")
(command "line" pt3 pt4 pt1 pt2 ""
         "arc" "" pt3          ; Arco contínuo
)
(command "layer" "m" "tm" "")
(command "line" pt5 pt6 ""
         "line" pt7 pt8 ""
)
)
```



# 3º Exemplo – Porta com Cotagem

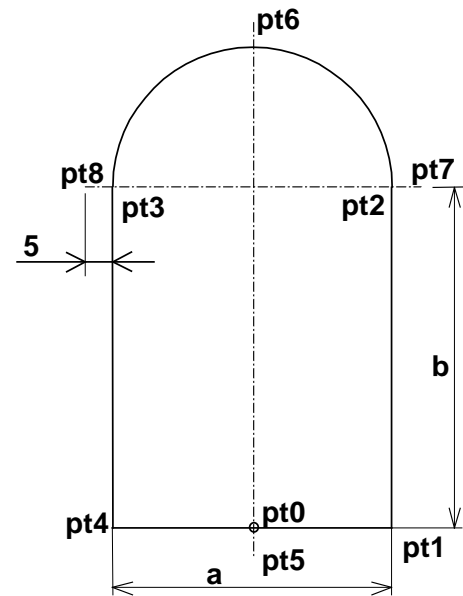
O mesmo programa do 2º exemplo, mas com a introdução de cotas.

```
;
; Programa para desenhar uma Porta de arco através de:
; Centro inferior, largura e altura direita. jof-92-11-25
;
(Defun c:porta()
  ; LEITURA
  ; vai ler os valores
  (setq a (getreal "\nLargura da porta      : "))
  (setq b (getreal "\nAltura direita da porta : "))
  (setq pt0 (getpoint "\nPonto Central inferior : "))
  ; CÁLCULOS
  ; vai calcular outros pontos necessários ao desenho
  (setq a2 (/ a 2))
```



# 3º Exemplo – Porta com Cotagem

```
(setq pi2 (/ pi 2))
(setq pt1 (polar pt0 0 a2))
(setq pt2 (polar pt1 pi2 b))
(setq pt4 (polar pt0 pi a2))
(setq pt3 (polar pt4 pi2 b))
(setq pt5 (polar pt0 (- pi2) 5))
(setq pt7 (polar pt2 0 5))
(setq pt8 (polar pt3 pi 5))
(setq pp (+ b a2 10))
(setq pt6 (polar pt5 pi2 pp))
;DESENHO
; Altera layer e desenha
(command "layer" "m" "TG" "")
(command "line" pt3 pt4 pt1 pt2 ""
          "arc" "" pt3          ; Arco contínuo
)
)
```





# 3º Exemplo – Porta com Cotagem

```
(command "layer" "m" "tm" "")
```

```
(command "line" pt5 pt6 ""
```

```
  "line" pt7 pt8 ""
```

```
)
```

```
; realização da cotagem
```

```
(command "layer" "m" "cotas" "")
```

```
(command "dim" "hor" pt4 pt1 (polar pt1 (- pi2) 10)
```

```
  (rtos (distance pt4 pt1) 2 0)
```

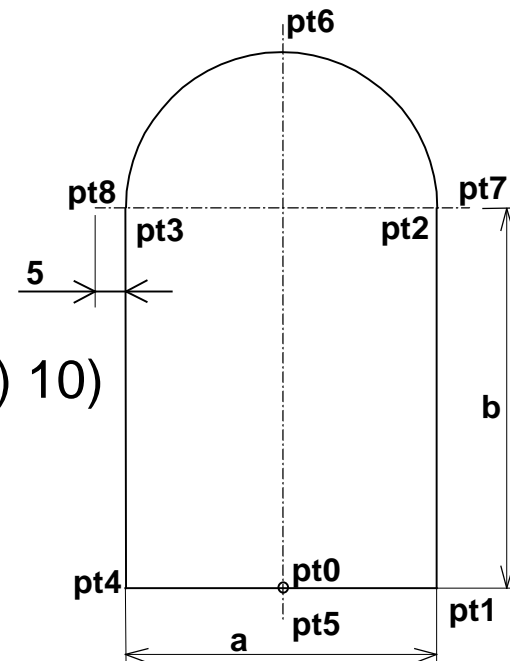
```
  "ver" pt1 pt2 (polar pt1 0 10)
```

```
  (rtos (distance pt1 pt2) 2 0)
```

```
  "exit"
```

```
)
```

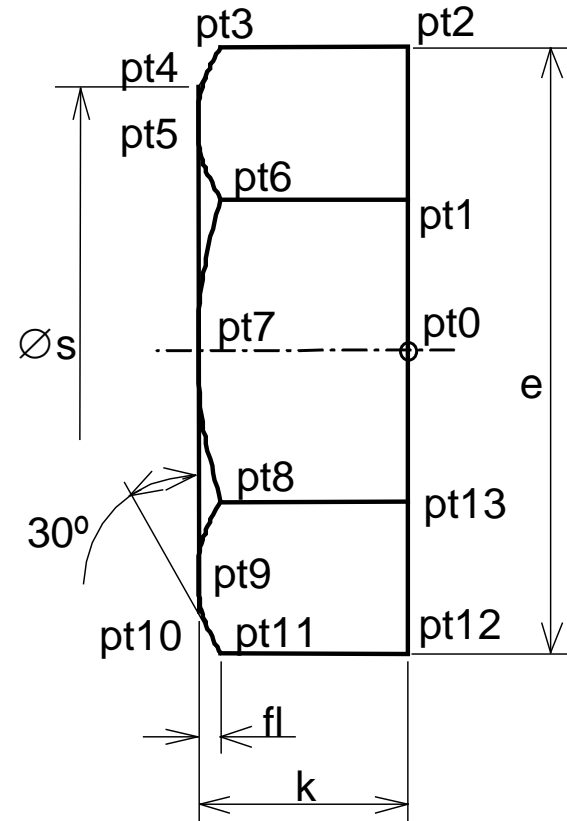
```
)
```



(2 – decimal, 0 –  
casas decimais)

# 4º Exemplo – Cabeça de Parafuso Hexagonal

Desenvolva um programa em AutoLISP que permita criar a figura junta (cabeça hexagonal dum parafuso), através da opção da vista pretendida, **ilog**. Tenha em atenção que será necessário definir o ponto **pt0**, a dimensão nominal da rosca **d**, o comprimento **L** e o ângulo da direcção do corpo do parafuso **alfa**. Os valores de **p**, **s**, **lmax** e **lmin** são lidos dum ficheiro, em que existirá uma linha com toda a informação necessária a cada diâmetro nominal.



# 4º Exemplo – Cabeça de Parafuso Hexagonal

Ficheiro a ser criado exteriormente (a primeira linha não pertence ao ficheiro e serve somente para ajudar a localizar os valores em cada linha).

## **parafhex.dat**

```
1234567890123456789012345678901234567890123456
3    ,0.5    ,5.5    , 2.0 ,6.    ,20.    ,30.    ,30
4    ,0.7    ,7.    , 2.8 ,8.    ,25.    ,40.    ,40
5    ,0.8    ,8.    , 3.5 ,10.   ,25.    ,50.    ,50
6    ,1.    ,10.   , 4.   ,12.   ,30.    ,60.    ,60
8    ,1.25   ,13.   , 5.3 ,16.   ,35.    ,80.    ,80
10   ,1.5    ,17.   , 6.4 ,20.   ,40.    ,100.   ,100
12   ,1.75   ,19.   , 7.5 ,25.   ,45.    ,100.   ,120
14   ,2.    ,22.   , 8.8 ,30.   ,50.    ,100.   ,140
16   ,2.    ,24.   ,10.   ,35.   ,55.    ,100.   ,160
20   ,2.5    ,30.   ,22.5 ,40.   ,65.    ,100.   ,200
24   ,3.    ,36.   ,15.   ,40.   ,80.    ,100.   ,240
30   ,3.5    ,46.   ,18.7 ,40.   ,90.    ,100.   ,300
36   ,4.    ,55.   ,22.5 ,40.   ,110.   ,100.   ,360
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
;
; Programa para desenho da cabeça do parafuso hexagonal
; alfa = 0 corresponde a desenhar corpo para a direita
(defun c:paraf()
  (setq oblip (getvar "blipmode") ; guarda modo de
                                ; colocação de cruces nos pontos
        oecho (getvar "cmdecho") ; guarda o modo de projecção dos
                                ; comandos
        olayer (getvar "clayer") ; guarda o layer corrente
        oosnap (getvar "osmode") ; guarda o modo osnap actual
  )
  (setvar "blipmode" 0) ; desliga a colocação de cruces nos pontos
  (setvar "cmdecho" 0) ; desliga a projecção dos comandos
  (setvar "osmode" 0) ; desliga qualquer modo osnap
  (initget 1) ; não admite resposta nula
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(setq pt0 (getpoint "\nDefina ponto de localização paraf : "))
(initget 1) ; não admite resposta nula
(setq alfa (getangle pt0 "\nDefina direcção do eixo : "))
(initget 1 "1 2 3")
(setq ilog (getkeyword "Três faces(1), Duas faces(2), de cima (3): "))
(if (/= ilog "3") ; pergunta pelo comprimento se não for
    ; a vista de cima
    (progn
        (initget (+ 1 2 4)) ; não permite que L seja vazio, nulo ou negativo
        (setq L (getreal "\nQual a comprimento do Parafuso :"))
    )
)
(setq linha nil ; inicializa os valores para controle
    listd nil
    i -1
)
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(while (= linha nil) ; while para leitura da tabela
  (setq f (open "parafhex.dat" "r")
        dif 0.
        ierro 0)
  (if (= f nil)
      (progn
        (prompt "\nFicheiro parafhex.dat não encontrado!")
        (setq lixo (getstring "\nIntroduza ENTER : ")
                ierro 1 ; ficheiro não existe
                linha "1")
        )
      )
  )
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(progn
  ; vai processar
  (initget (+ 1 2 4)) ; não permite que d seja vazio, nulo ou negativo
  (setq d (getreal "\nQual a dimensão do Parafuso - M...- :"))
  (while (and (/= d dif) (= ierro 0))
    (setq linha (read-line f))
    (if (and (/= linha nil) (= ierro 0)) ; então
      (setq dif (atof (substr linha 1 4)))
      listd (append listd (list dif)) ; guarda diâmetro lido
      i (+ i 1)
    )
    (setq dif d ; senão
      ierro 2 ; diâmetro não encontrado
    )
  )
)
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(close f)
(if (= ierro 2)
    (progn
        ; então
        (setq j 0)
        (prompt "\nDiametro ERRADO. No ficheiro existem: ")
        (while (<= j i) ; escreve os diâmetros lidos do ficheiro
            (princ (nth j listd)) ; escreve elemento j da lista listd
            (princ " , ")
            (setq j (+ j 1))
        )
        (setq listd nil
            i -1
            dif 0.0
        )
    )
) ; fim do if do ierro 2
```



# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
) ; fim do progn de processamento
) ; Fim do if f=nil
) ; Fim do while para leitura
(if (= ierro 0) ; no caso de ter encontrado o d na tabela, desenha
  (progn ; então1
        ; define os restantes parâmetros
        (setq p (atof (substr linha 6 5))) ; passo da rosca
        (setq s (atof (substr linha 12 5))) ; entre-faces do sextavado
        (setq k (atof (substr linha 18 4)))
                ; altura da cabeça do parafuso
        (setq lmin1 (atof (substr linha 24 5)))
                ; compr mínimo para todo roscado
        (setq lmin2 (atof (substr linha 30 4)))
                ; compr mínimo para o com liso
        (setq lmax1 (atof (substr linha 36 5))) ;compr máx p/ todo roscado
        (setq lmax2 (atof (substr linha 42 4))) ; compr máx para com liso
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(if (/= ilog "3") ;controla o comprimento se não for a vista de cima
  (progn
    ; então2
    (if (< L lmax2)
      (if (> L lmin1) ; então3
        (setq op 1) ; então4
        (setq op 3) ; senão4
      )
      (setq op 2) ; senão3
    )
    (if (equal op 2)
      (prompt "\nComprimento demasiado longo.")
      (if (equal op 3)
        (prompt "\nComprimento demasiado curto.")
      )
    )
  )
)
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(setq op 1) ; senão2 define um valor para o caso da vista de cima
)
(if (equal op 1) ; no caso de ser aceitável o comprimento
    (progn ; então2
        (setq pi6 (/ pi 6))
        (setq pi2 (/ pi 2))
        (setq e (/ s (cos pi6)))
        ; definição de layer
        (command "layer" "m" "tg" "")
        (setq fl (/ (- e s) 2))
        (setq fl (* fl (/ (sin pi6) (cos pi6))))
        (if (= ilog "1")
            (progn ; então3 definição de pontos
```

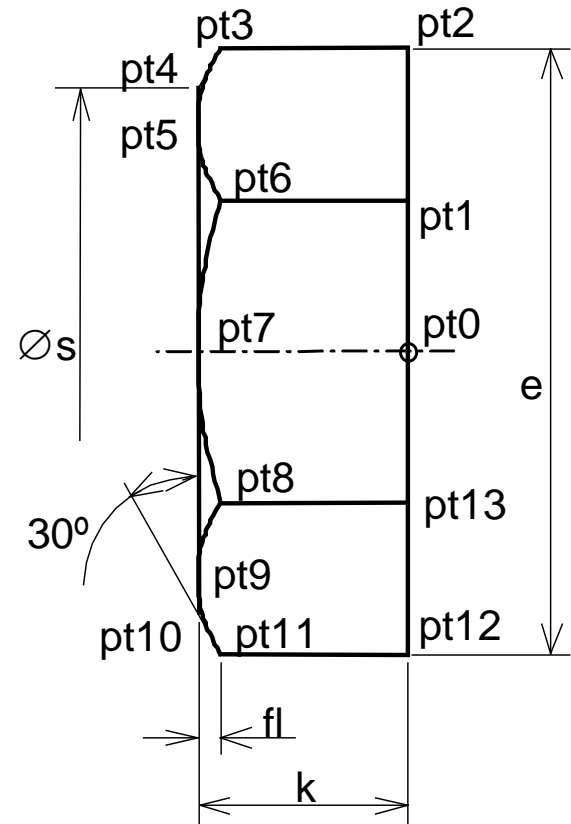
# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(setq pt1 (polar pt0 pi2 (/ e 4)))  
(setq pt2 (polar pt0 pi2 (/ e 2)))  
(setq pt3 (polar pt2 pi (- k fl)))  
(setq pt7 (polar pt0 pi k))  
(setq pt4 (polar pt7 pi2 (/ s 2)))  
(setq pt5 (polar pt7 pi2 (/ (* e 3) 8)))  
(setq pt6 (polar pt1 pi (- k fl)))  
(setq pt12 (polar pt0 (- pi2) (/ e 2)))  
(setq pt13 (polar pt0 (- pi2) (/ e 4)))  
(setq pt8 (polar pt13 pi (- k fl)))  
(setq pt9 (polar pt7 (- pi2) (/ (* e 3) 8)))  
(setq pt10 (polar pt7 (- pi2) (/ s 2)))  
(setq pt11 (polar pt12 pi (- k fl)))
```

; vai agora desenhar parafuso

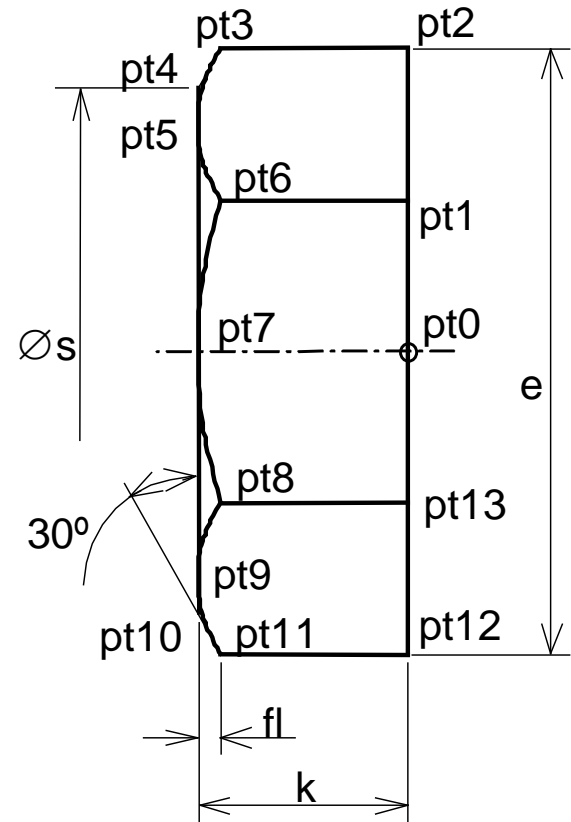
# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(command "arc"  
  (rot pt3)  
  (rot pt5)  
  (rot pt6)  
  "arc"  
  (rot pt6)  
  (rot pt7)  
  (rot pt8)  
  "arc"  
  (rot pt8)  
  (rot pt9)  
  (rot pt11)  
  "line"
```



# 4º Exemplo – Cabeça de Parafuso Hexagonal

(rot pt12)  
(rot pt2)  
(rot pt3)  
(rot pt4)  
(rot pt10)  
(rot pt11)  
"c"  
"line"  
(rot pt1)  
(rot pt6)  
""  
"line"  
(rot pt13)  
(rot pt8)



# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
      "" ; falta corpo
    )
  )
  (if (= ilog "2") ; senão3 ilog 2 - parafuso com duas faces
      (prompt "\nAinda não implementado!") ; então4
      (progn ; senão4 ilog 3 - desenho do parafuso em outra vista
          (setq pt1 (polar pt0 alfa (/ e 2))
                pt2 (polar pt0 (+ (/ pi 3) alfa) (/ e 2))
          )
      )
  )
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

```
(command "zoom" "a")  
(command "line"  
  pt1  
  pt2  
  ""  
  "array"  
  (entlast)  
  ""  
  "p"  
  pt0  
  6  
  360  
  ""  
  "circle"
```



# 4º Exemplo – Cabeça de Parafuso Hexagonal

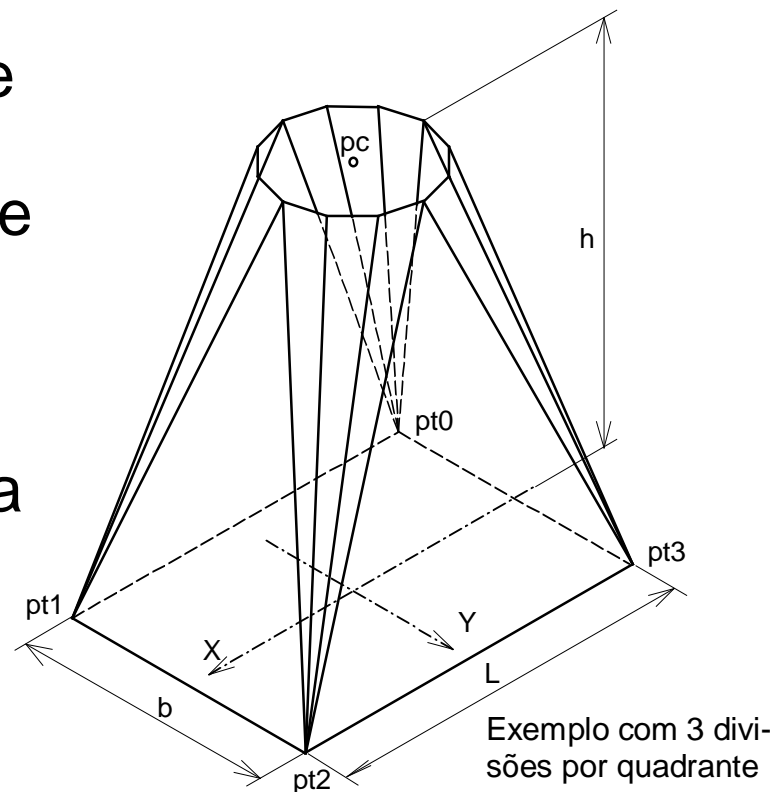
```
      pt0
      (/ s 2)
    )
  )
)
)
)
)
)
)
(prompt "\nO diâmetro introduzido não existe no ficheiro!")
; senão1
)
; fim do if de ierro = 0
(command "layer" "m" olay "") ; repor estado anterior
(setvar "osmode" oosnap)
```

# 4º Exemplo – Cabeça de Parafuso Hexagonal

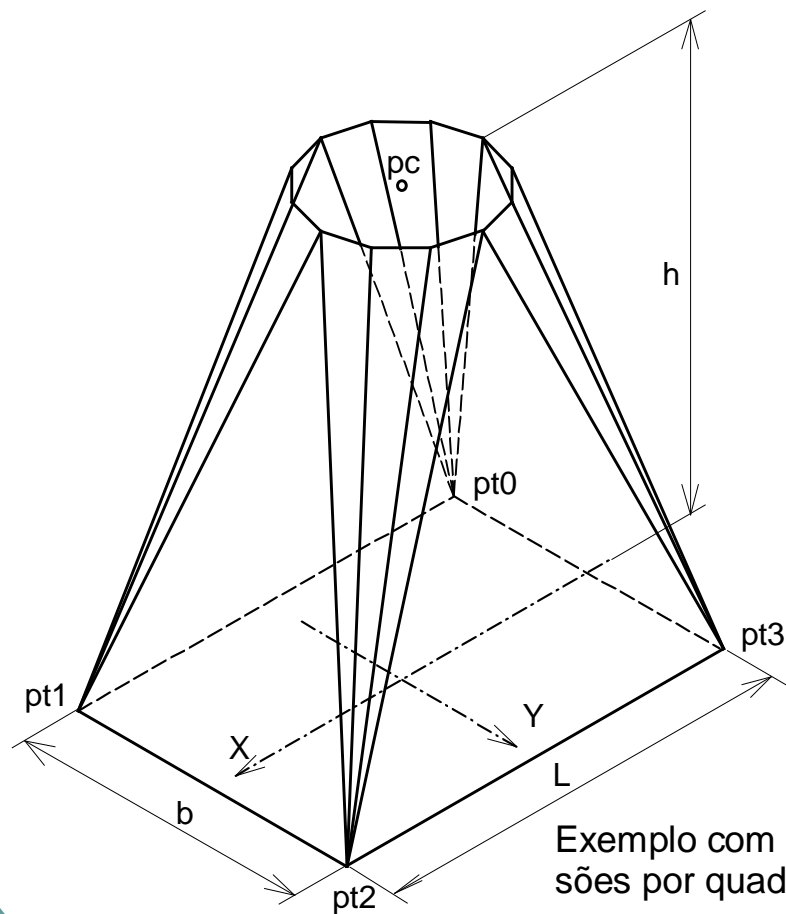
```
(setvar "blipmode" oblip)
(setvar "cmdecho" oecho)
)
; Definição da subrotina de rotação em torno de pt0 e de um ângulo
; alfa
(defun rot (ptc / dx dy seno cosse xx yy ptn)
  (setq dx  (- (car ptc) (car pt0))
        dy  (- (cadr ptc) (cadr pt0))
        seno (sin alfa)
        cosse (cos alfa)
        xx  (- (* cosse dx) (* seno dy))
        yy  (+ (* seno dx) (* cosse dy))
        ptn (list (+ xx (car pt0)) (+ yy (cadr pt0))))
  )
)
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

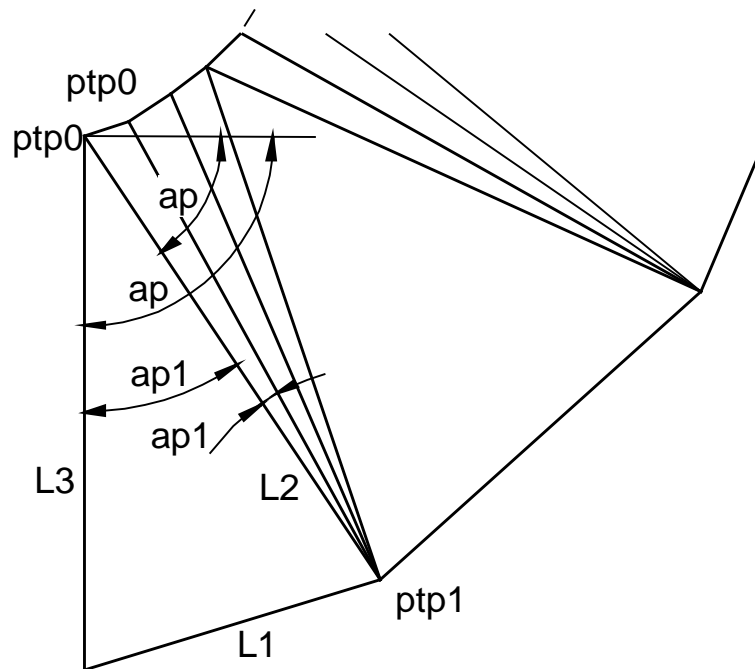
Exemplo de um programa em AutoLISP, para geração e planificação de uma superfície de ligação entre um tubo de secção quadrada e um tubo de secção circular, com os seus eixos na direcção de Z e concêntricos. A geração é obtida pela triangularização da respectiva superfície, como é tecnicamente proposto em termos de planificação.



# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular



Exemplo com 3 divi-  
sões por quadrante



# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(Defun C:Chamin()  
  (setvar "blipmode" 0)  
  (setvar "cmdecho" 0)  
  (setvar "osmode" 0)  
  ; Leitura das grandezas significativas  
  (initget (+ 1 2))  
  (setq h (getreal "\nAltura: "))  
  (initget (+ 1 2 4))  
  (setq b (getreal "\nLargura da chaminé: "))  
  (initget (+ 1 2 4))  
  (setq l (getreal "\nComprimento da chaminé: "))  
  (initget (+ 1 2 4))  
  (setq d (getreal "\nDiâmetro da saída: "))  
  (initget (+ 1 2 4))
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(setq n (getint "\nNº de divisão por quadrante: "))  
; Localização das figuras  
(initget 1)  
(setq pt0 (getpoint "\nPonto de fixação da representação 3D: "))  
(initget 1)  
(setq ptp0 (getpoint "\nPonto de fixação da Planificação 2D: "))  
; Cálculo de pontos particulares (centro da saída circular e cantos do  
; rectângulo)  
(setq pc (list (+ (car pt0) (/ l 2))  
               (+ (cadr pt0) (/ b 2))  
               (+ (caddr pt0) h)  
               )  
)  
(setq pt1 (polar pt0 0 l))
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(setq pt2 (polar pt1 (/ pi 2) b))  
(setq pt3 (polar pt0 (/ pi 2) b))  
(setq i 1) ; contador de quadrantes  
(setq da (/ pi 2 n)) ; incremento angular  
(setq a 0.0) ; ângulo inicial  
(setq r (/ d 2))  
(setq ap (- (/ pi 2))) ; planificação (ângulo inicial  
; considerado de -90º)  
  
(while (<= i 4) ; ciclo while sobre os quatro quadrantes  
  (if (= i 1) ; inicialização de cada quadrante  
    (progn  
      (setq pt0c pt1)  
      (setq ptc pt2)
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
)  
)  
(if (= i 2)  
  (progn  
    (setq pt0c pt2)  
    (setq ptc pt3)  
  )  
)  
(if (= i 3)  
  (progn  
    (setq pt0c pt3)  
    (setq ptc pt0)  
  )  
)  
)
```



# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(if (= i 4)
  (progn
    (setq pt0c pt0)
    (setq ptc pt1)
  )
)
; 1º do triângulo de bico para cima
(setq ptcc0 (polar pc a r))
(command "3dface" pt0c ptc ptcc0 "" "")
(setq l1 (distance pt0c ptc))
(setq l2 (distance ptc ptcc0))
(setq l3 (distance pt0c ptcc0))
(setq ap1 (alftri l1 l2 l3)) ; alftri função definida no fim
; Define triângulo de bico para cima na planificação em XY
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(command "3dface"  
  ptp0  
  (polar ptp0 ap l3)  
  (polar ptp0 (+ ap ap1) l2)  
  ""  
  ""  
)  
(setq ap (+ ap ap1))  
(setq j 1) ; contador de divisões do quadrante  
(while (<= j n) ; ciclo sobre cada quadrante para  
  ; triangularizar  
  (setq a (+ a da))  
  (setq ptcc1 (polar pc a r))
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(command "3dface" ptcc0 ptc ptcc1 "" "")  
; Define triângulo no espaço  
  (setq l1 (distance ptcc0 ptcc1))  
  (setq l2 (distance ptcc0 ptc))  
  (setq l3 (distance ptcc1 ptc))  
  (setq ap1 (alftri l1 l2 l3))  
  (setq ptp1 (polar ptp0 ap l2))  
; Desenha triângulo na planificação em XY  
  (command "3dface"  
    ptp0  
    ptp1  
    (polar ptp1 (- ap ap1 pi) l3)  
    ""  
    ""  
  )
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

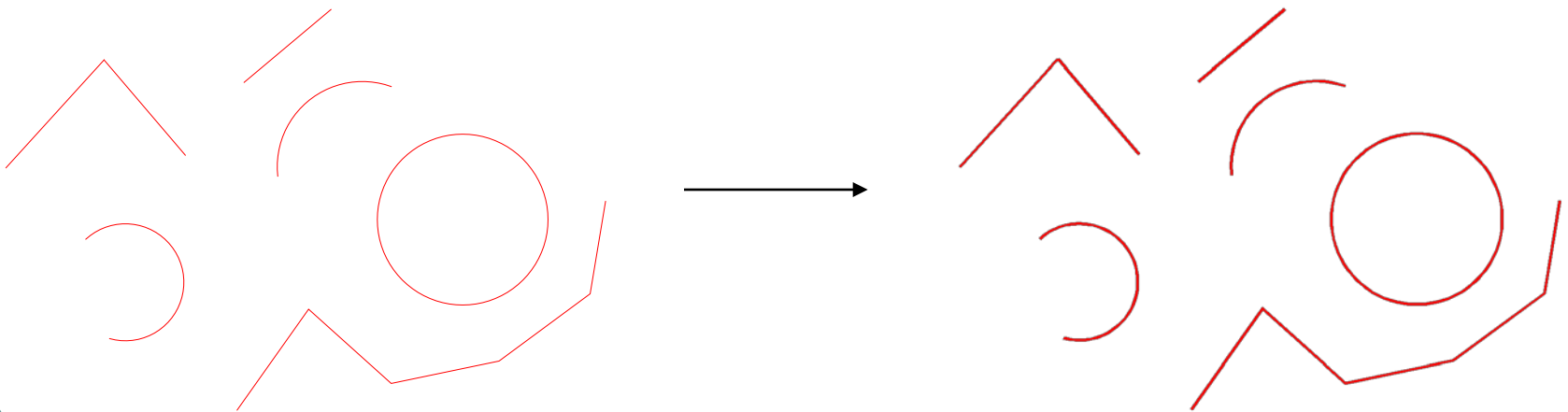
```
(setq ptp0 (polar ptp1 (- ap ap1 pi) l3))
(setq ap (- ap ap1))
(setq ptcc0 ptcc1)
(setq j (+ j 1))
) ; Fim do while em j
(setq i (+ i 1))
) ; Fim do while em i
(setvar "blipmode" 1)
) ; Fim do programa
; Executa a função ARCOS (arcocosseno)
(defun Arcos (p / tg)
  (if (= (abs p) 1)
    (setq tg (/ (* pi (- 1 p)) 2))
    (setq tg (- (/ pi 2) (atan (/ p (sqrt (- 1 (* p p))))))))
)
```

# 5º Exemplo – Planificação de Superfície ligação Tubo Redondo/Rectangular

```
(setq tg tg)
)
; Aplica a lei dos cossenos a um triângulo
(defun Alftri (l1 l2 l3 / p a)
  (setq p (/ (- (+ (* l2 l2) (* l3 l3)) (* l1 l1)) l2 l3 2))
  (setq a (arcos p)) ; arcos função definida anteriormente
)
```

# 6º Exemplo – Engrossamento de linhas do LAYER TG

Programa para engrossamento do traço das entidades PLINE, LINE, ARC e CIRCLE, transformando-as em PLINE com uma determinada espessura.



# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
; Programa para transformação de entidades "LINE", "ARC" e  
; "CIRCLE" associadas ao layer TLAYER, em PLINE com espessura  
; TESP e alterar a espessura de qualquer POLYLINE em TLAYER  
; para TESP
```

```
; Joaquim O. Fonseca - DEMEGI/GDI - 1993/11/10
```

```
*****
```

```
;
```

```
;
```

```
;
```

```
(defun c:espestg()
```

```
;
```

```
;
```

```
; guarda valores de variáveis de ambiente
```

```
;
```

```
;
```

```
(setq olay (getvar "clayer"))
```

```
(setq odecho (getvar "cmdecho"))
```

# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
(setvar "cmdecho" 0)
(setvar "blipmode" 0)
(command "osnap" "none")           ; anula qualquer modo
                                   ; OSNAP permanente
;
; define o layer a alterar e qual a espessura
;
(setq tlayer "TG")
(setq tesp 0.4)
;
; Processa primeiro POLYLINE, senão repetiria o processamento
; das outras entidades, entretanto transformadas em POLYLINE
; cria lista com todas as entidades do desenho que satisfazem o
; "filtro" definido 8=apontador do "layer" e 0=apontador do tipo
```



# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
(setq ss1 (ssget "X" (list (cons 8 tlayer) (cons 0 "lwpolyline"))))
(if (/= ss1 nil)
  (progn
    (setq ne (sslenght ss1))
    (setq i 0)
    (while (< i ne)
      (setq le (ssname ss1 i))
      (command "pedit" le "w" tesp "")
      (setq i (+ i 1)))
    )
  )
)
```

# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
;
; Processa LINE
;
(setq ss1 (ssget "X" (list (cons 8 tlayer) (cons 0 "line"))))
(if (/= ss1 nil)
  (progn
    (setq ne (sslength ss1))
    (setq i 0)
    (while (< i ne)
      (setq le (ssname ss1 i))
      (command "pedit" le "" "w" tesp "")
      (setq i (+ i 1)))
    )
  )
)
```

# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
;
; Processa ARC
;
(setq ss1 (ssget "X" (list (cons 8 tlayer) (cons 0 "arc"))))
(if (/= ss1 nil)
  (progn
    (setq ne (sslength ss1))
    (setq i 0)
    (while (< i ne)
      (setq le (ssname ss1 i))
      (command "pedit" le "" "w" tesp "")
      (setq i (+ i 1)))
    )
  )
)
```

# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
;
; Processa CIRCLE dividindo-o em dois arcos de POLYLINE
;
(setq ss1 (ssget "X" (list (cons 8 tlayer) (cons 0 "circle"))))
(if (/= ss1 nil)
    (progn
      (setq ne (sslenght ss1))
      (setq i 0)
      (command "layer" "s" tlayer "")
      (while (< i ne)
        (setq le (ssname ss1 i))
        (setq pt0 (cdr (assoc 10 (entget le)))) ; centro
        (setq r (cdr (assoc 40 (entget le)))) ; raio
        (command "erase" le ""
          "pline" (polar pt0 0 r) "w" tresp "" "a" "ce" pt0 (polar pt0 pi r) "cl"
        )
      )
    )
  )
```

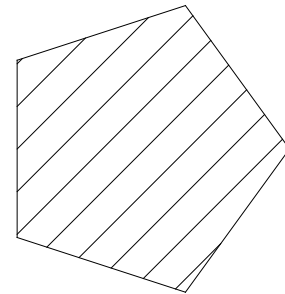
# 6º Exemplo – Engrossamento de linhas do LAYER TG

```
        (setq i (+ i 1))
      )
    )
  )
  (command "layer" "m" olay "")
  (setvar "blipmode" 1)
  (setvar "cmdecho" odecho)
)
```

# 7º Exemplo – Polígono Regular e tracejado do seu interior

Programa para construir um polígono regular com **n** número de lados e posteriormente o tracejado no seu interior.

```
(Defun C:poligono()  
  (initget (+ 1 2 4))  
  (setq n (getint "\nNº de lados: "))  
  (initget (+ 1 2 4))  
  (setq r (getreal "Raio do círculo: "))  
  (setq pt0 (getpoint "Centro do polígono: "))  
  (setq ang (/ (* pi 2) n))  
  (command "LINE" (polar pt0 0 r) (polar pt0 ang r) ""))
```



# 7º Exemplo – Polígono Regular e tracejado do seu interior

```
(setq e1 (entlast))           ; guarda a primeira entidade a
                              ; processar no HATCH
(command "ARRAY" "I" "" "p" pt0 n 360 "y")
(command "HATCH" "u" 45 (/ r 5) "n")
(while (/= e1 nil)           ; introduz no comando todas as ;
                              ; entidades a partir
    (command e1)             ; de e1 até à última desenhada
    (setq e1 (entnext e1))   ; OBS.: e1 vai tomando o nome das
)                             ; sucessivas entidades
(command "")                 ; “fecha” o comando HATCH
)
```

