



# Capítulo 24

## Soluciones

Debe entenderse que estas no son LAS soluciones a los problemas, sino UNAS soluciones a los problemas. Existen muchas formas de plantear programas para resolver un mismo problema. Ten esto en cuenta cuando leas estas soluciones.

### 24.1. Ejercicios 3.2

Aunque el capítulo 3 no pide dibujar con primitivas ni diseñar procedimientos, quizás alguien tenga interés en cómo se hicieron los dibujos.

Los Archivos `rana.lgo`, `robot.lgo`, `barco.lgo`, `horreo.lgo`, `piramides.lgo` y `Torre_Hercules.lgo` pueden descargarse desde la página del curso.

Para dibujar, simplemente hacer *clic* en el botón de **Comando de inicio**

### 24.2. Ejercicios 4.3

1. Cuadrado en sentido antihorario

```
avanza 150 giraizquierda 90 avanza 150 giraizquierda 90
avanza 150 giraizquierda 90 avanza 150
```

2. Cuadrado en sentido horario

```
avanza 150 giraderecha 90 avanza 150 giraderecha 90
avanza 150 giraderecha 90 avanza 150
```

3. Triángulo en sentido antihorario

```
giraderecha 90
avanza 150 giraizquierda 120 avanza 150 giraizquierda 120 avanza 150
```

## 4. Rectángulo (en sentido horario)

avanza 100 giraderecha 90 avanza 200 giraderecha 90  
 avanza 100 giraderecha 90 avanza 200

## 5. Hexágono regular (en sentido horario)

avanza 100 giraderecha 60 avanza 100 giraderecha 60  
 avanza 100 giraderecha 60 avanza 100 giraderecha 60  
 avanza 100 giraderecha 60 avanza 100

## 6. a) "Letra C"

avanza 300 giraderecha 90    avanza 200 giraderecha 90  
 avanza 100 giraderecha 90    avanza 100 giraizquierda 90  
 avanza 100 giraizquierda 90    avanza 100 giraderecha 90  
 avanza 100 giraderecha 90    avanza 200

## b) Flecha inclinada

giraderecha 45    avanza 120 giraderecha 90    avanza 40  
 giraizquierda 135    avanza 120 giraizquierda 90    avanza 120  
 giraizquierda 135    avanza 40 giraderecha 90    avanza 120  
 giraizquierda 90    avanza 90

## c) "Diábolo"

giraderecha 45    avanza 283.85 giraizquierda 135    avanza 200  
 giraizquierda 135    avanza 283.85 giraderecha 135    avanza 200

## d) Camiseta

avanza 200 giraizquierda 135    avanza 120 giraderecha 135  
 avanza 80 giraderecha 45    avanza 120 giraderecha 45  
 avanza 200 giraderecha 45    avanza 120 giraderecha 45  
 avanza 80 giraderecha 135    avanza 120 giraizquierda 135  
 avanza 200 giraderecha 90    avanza 200

## e) Fila de cuadrados

avanza 50 giraderecha 90    avanza 100 giraizquierda 90  
 avanza 100 giraderecha 90    avanza 100 giraizquierda 90  
 avanza 100 giraderecha 90    avanza 50 giraderecha 90  
 avanza 50 giraderecha 90    avanza 100 giraizquierda 90  
 avanza 100 giraderecha 90    avanza 100 giraizquierda 90  
 avanza 100 giraderecha 90    avanza 50

## f) Proyección de un cubo

```

avanza 200 giraderecha 90 avanza 200 giraderecha 90
avanza 200 giraderecha 90 avanza 200 giraderecha 90
avanza 200 giraderecha 45 avanza 70.7 giraderecha 45
avanza 200 giraderecha 135 avanza 70.7 retrocede 70.7
giraizquierda 45 avanza 200 giraderecha 45 avanza 70.7

```

### 24.3. Ejercicios 4.5

1. Cuadrado:

```

repite 4 [ avanza 150 giraizquierda 90 ]

```

2. Triángulo equilátero:

```

giraderecha 90
repite 3 [ avanza 150 giraizquierda 120 ]

```

3. Hexágono regular:

```

repite 6 [ avanza 100 giraderecha 60 ]

```

4. Cuadrado centrado:

```

subelapiz
retrocede 100 giraizquierda 90 avanza 100 giraderecha 90
bajalapiz
repite 4 [ avanza 200 giraderecha 90 ]

```

5. Rectángulo relleno:

```

repite 2 [avanza 100 giraderecha 90 avanza 200 giraderecha 90]
subelapiz
giraderecha 45 avanza 30 giraizquierda 45
bajalapiz
rellenazona

```

6. Hexágono con diagonales:

```

repite 6 [ repite 3 [ avanza 100 giraderecha 120 ] giraderecha 60]

```

7. Radios de una rueda:

```

repite 24 [ avanza 150 retrocede 150 giraderecha (360/24) ]

```

8. a) Escalera:

```
repite 5 [ avanza 50 giraderecha 90 avanza 50 giraizquierda 90 ]
retrocede 250 giraizquierda 90 avanza 250
```

b) Cuadrados dentro de cuadrado:

```
repite 4 [ avanza 250 giraderecha 90
           repite 4 [ avanza 75 giraderecha 90 ] ]
```

c) Camino Cortado:

```
giraderecha 90   avanza 100 subelapiz
giraderecha 90   avanza  50 bajalapiz
giraizquierda 90 avanza 100 subelapiz
giraizquierda 90 avanza  50 bajalapiz
giraderecha 90   avanza 100
```

d) Barras verticales:

```
repite 6 [ avanza 250 subelapiz retrocede 250 giraderecha 90
           avanza 50 giraizquierda 90 bajalapiz ]
```

e) Barras horizontales:

```
giraderecha 90
repite 6 [ avanza 250 subelapiz retrocede 250 giraizquierda 90
           avanza  50 giraderecha 90 bajalapiz ]
```

f) Barras oblicuas:

```
giraderecha 45
repite 6 [ avanza 250 subelapiz retrocede 250 giraderecha 45
           avanza  50 giraizquierda 45 bajalapiz ]
```

## 24.4. Ejercicios 5.2

1. Triángulo equilátero:

```
para triangulo
  giraizquierda 90
  repite 3 [ avanza 150 giraderecha 120 ]
fin
```

2. Hexágono regular:

```
para hexagono
  repite 6 [ avanza 150 giraderecha 60 ]
fin
```

## 3. Cuadrado centrado:

```

para centrocuadrado
  subelapiz
    retrocede 100 giraizquierda 90 avanza 100 giraderecha 90
  bajalapiz
    repite 4 [ avanza 200 giraderecha 90 ]
fin

```

## 4. Radios de una rueda:

```

para 24radios
  repite 24 [ avanza 150 retrocede 150 giraderecha (360/24) ]
fin

```

## 5. a) Molinillo:

```

  repite 4 [ rectangulo giraderecha 90 ]

```

## b) Doble escalera (Nota: Sólo para el cuadrado definido como primer ejemplo)

```

  repite 5 [ cuadrado giraderecha 90 avanza 100 giraizquierda 90 ]
  subelapiz
    giraizquierda 90 avanza 400 giraderecha 90 avanza 100
  bajalapiz
    repite 3 [ cuadrado giraderecha 90 avanza 100 giraizquierda 90 ]
  subelapiz
    giraizquierda 90 avanza 200 giraderecha 90 avanza 100
  bajalapiz
  cuadrado

```

## 6. ■ Rombo:

```

para rombo
  repite 2 [ avanza 50 giraderecha 30 avanza 50 giraderecha 150 ]
fin

```

■ Hélice: repite 9 [ rombo giraizquierda (360/9) ]

**24.5. Ejercicios 5.4**

## 1. para girados

```

  repite 6 [ hexagono giraderecha 60 ]
fin

```

2. para hiper\_cuadrado
 

```

      repite 4 [
        repite 4 [ cuadrado avanza 50 ]
        giraderecha 90 avanza 50 ]
      fin
      
```
3. para casi\_cuadrado
 

```

      repite 3 [avanza 100 giraderecha 90 ]
      fin
      
```

  - a) para separados
 

```

          giraizquierda 90
          repite 2 [ casi_cuadrado
            subelapiz giraizquierda 90 avanza 100 bajalapiz ]
          fin
          
```
  - b) para podium
 

```

          repite 3 [ giraizquierda 90 casi_cuadrado giraizquierda 90 ]
          avanza 100
          fin
          
```
  - c) para cruz\_griega
 

```

          repite 4 [ giraizquierda 90 casi_cuadrado giraizquierda 90 ]
          fin
          
```
4. El patrón es:

```

para cenefa
  avanza 100 cuadrado
fin

```

- a) para cenefa\_serie
 

```

      repite 3 [
        cenefa giraizquierda 90 avanza 50 giraderecha 90 ]
      fin
      
```
- b) para cenefa2
 

```

      repite 4 [
        cenefa giraizquierda 90 ]
      fin
      
```
- c) para cenefa3
 

```

      repite 4 [
        cenefa retrocede 50 giraizquierda 90 retrocede 50 ]
      fin
      
```

```

d) para cenefa4
    repite 4 [
        cenefa giraderecha 90 ]
    fin

5. para fila_trian
    giraderecha 90
    repite 5
    [ triangulo avanza 100 ]
    fin

para triangulo
    repite 3
    [ avanza 100 giraizquierda 120 ]
    fin

```

## 24.6. Ejercicios 6.3

```

1. para triangulo :lado
    repite 3 [ avanza :lado giraizquierda 120 ]
    fin

2. para rueda :radio
    repite 36 [ avanza :radio retrocede :radio giraderecha 10 ]
    fin

3. para bandera :mastil :alto :ancho
    avanza :mastil retrocede :alto
    rectangulo :alto :ancho
    fin

4. para poligono :n :largo
    haz "angulo 360/:n
    repite :n
    [ avanza :largo giraizquierda :largo ]
    fin

5. para ruedas :n :largo :distancia
    repite :n [ rueda :largo
        subelapiz
        giraizquierda 90 avanza :distancia giraderecha 90
        bajalapiz ]
    fin

(Recuerda que rueda ya se diseñó antes)

```

## 24.7. Ejercicios 7.4

1. a) para cuadrados\_1 :lado
 

```

      repite 4 [ cuadrado :lado
        subelapiz
        giraderecha 90 avanza (3*:lado) giraizquierda 90 avanza (2*:lado)
        giraizquierda 90 bajalapiz ]
      subelapiz
      giraizquierda 90 avanza :lado giraderecha 90 avanza :lado
      bajalapiz
      cuadrado (3*:lado)
      fin
      
```
- b) para cuadrados\_2 :lado
 

```

      haz "separacion :lado/10
      repite 6 [ cuadrado :lado
        haz "lado :lado - 2*:separacion
        subelapiz
        giraderecha 90 avanza :separacion
        giraizquierda 90 avanza :separacion
        bajalapiz ]
      fin
      
```
2. para baldosas :n :lado
 

```

      subelapiz
      giraizquierda 90 avanza (:n*:lado)/2
      giraderecha 90 retrocede (:lado/2)
      bajalapiz
      repite :n [ cuadrado :lado
        giraderecha 90 avanza :lado giraizquierda 90 ]
      fin
      
```
3. a) para barras\_menguantes
 

```

      haz "largo 250
      repite 6 [
        avanza :largo
        subelapiz
        retrocede :largo giraderecha 90
        avanza 50 giraizquierda 90
        bajalapiz
        haz "largo :largo - 50 ]
      fin
      
```
- b) para triangulos
 

```

      giraderecha 90
      
```



- ```

    haz "lado 300
    repite 3 [
        repite 3 [
            avanza :lado giraizquierda 120 ]
        haz "lado :lado - 100
        avanza 50 ]
    fin

```
4. para division\_entera :Dvdo :dsor
 

```

        escribe frase [El dividendo es:] :Dvdo
        escribe frase [El divisor es:] :Ddor
        haz "Cociente truncar (:Dvdo/:dsor)
        escribe frase [El cociente es:] :cociente
        haz "resto :Dvdo - :dsor * :cociente
        escribe frase [El resto es:] :resto
      fin

```
  5. para Pitagoras :cateto1 :cateto2
 

```

        haz "hipotenusa raizcuadrada (suma (potencia :cateto1 2)
            (potencia :cateto2 2))
        escribe frase [La hipotenusa mide:] :hipotenusa
      fin

```
  6. para radios :n :largo
 

```

        repite :n
        [ avanza :largo retrocede :largo giraizquierda (360/:n) ]
      fin

```
  7. para poligono\_regular :n :lado
 

```

        repite :n
        [ avanza :lado giraizquierda (360/:n) ]
      fin

```
  8. para suerte
 

```

        escribe 20 + 5 * azar 7
      fin

```
  9. para dados
 

```

        haz "dato1 1 + azar 6
        haz "dato2 1 + azar 6
        escribe suma :dato1 :dato2
      fin

```
  10. para poten :x :n
 

```

        haz "resultado 1

```

```

    repite :n
      [ haz "resultado :resultado * :x ]
    escribe :resultado
  fin

```

```

11. para factorial :n
    haz "resultado 1
    repite :n
      [ haz "resultado :resultado * contador ]
    escribe :resultado
  fin

```

```

12. para suma_potencias :n
    haz "resultado 0
    repite :n
      [ haz "resultado :resultado + (potencia 2 contador) ]
    escribe :resultado
  fin

```

## 24.8. Ejercicios 8.3

1. para rectangulo :largo :alto
 

```

        subelapiz
        ponx (-:largo/2) pony (-:alto/2)
        bajalapiz
        ponx (:largo/2) pony (:alto/2)
        ponx (-:largo/2) pony (-:alto/2)
      fin

```
2. para isosceles :cateto
 

```

        ponposicion lista :cateto 0
        ponposicion lista :cateto :cateto
        ponposicion [ 0 0 ]
      fin

```
3. para segmento :x1 :y1 :x2 :y2
 

```

        subelapiz
        ponposicion lista :x1 :y1
        bajalapiz
        ponposicion lista :x2 :y2
      fin

```
4. para cuadrilatero :x1 :y1 :x2 :y2 :x3 :y3 :x4 :y4
 

```

        subelapiz

```

```

ponposicion lista :x1 :y1
bajalapiz
ponposicion lista :x2 :y2
ponposicion lista :x3 :y3
ponposicion lista :x4 :y4
ponposicion lista :x1 :y1
fin

```

```

5. para dist_ptos :x1 :y1 :x2 :y2
    segmento :x1 :y1 :x2 :y2
    subelapiz
    ponposicion lista (:x1 + :x2)/2 (:y1 + :y2)/2
    giraderecha 90 avanza 100 giraizquierda 90
    bajalapiz
    haz "distancia (potencia (:x1 - :x2) 2) + (potencia (:y1 - :y2) 2)
    haz "distancia raizcuadrada :distancia
    rotula (frase [La distancia entre los puntos es: ] :distancia "pasos)
fin

```

```

6. para norte
    ponrumbo 0
fin
para sur
    ponrumbo 180
fin
para este
    ponrumbo 90
fin
para oeste
    ponrumbo -90
fin
para nordeste
    ponrumbo 45
fin
para noroeste
    ponrumbo -45
fin
para sudeste
    ponrumbo 135
fin
para suroeste
    ponrumbo 215
fin

```

- ```

7. para triangulo :x1 :y1 :x2 :y2 :x3 :y3
  subelapiz
  ponposicion lista :x1 :y1
  bajalapiz
  ponposicion lista :x2 :y2
  ponposicion lista :x3 :y3
  ponposicion lista :x1 :y1
  subelapiz
  ponposicion lista (:x1+:x2+:x3)/3 (:y1+:y2+:y3)/3
  bajalapiz rellenazona
fin

8. a) para cuadrado_marcos
  haz "n 1
  haz "paso 25
  haz "salto 5
  repite 2 [
    repite 4 [
      avanza :paso
      subelapiz
      avanza :salto * :paso
      bajalapiz
      avanza :paso giraderecha 90 ]
    haz "salto :salto - 2
    subelapiz ponxy (:n * :paso) (:n * :paso)
    haz "n :n +1
    bajalapiz ]
  repite 4 [
    avanza 3 * :paso giraderecha 90 ]
fin

b) para enfrentadas
  haz "angulo 90
  haz "paso 50
  repite 2 [
    avanza :paso giraderecha :angulo
    avanza (2*:paso) giraizquierda :angulo
    avanza (3*:paso) giraizquierda :angulo
    avanza (2*:paso) giraderecha :angulo
    avanza :paso giraderecha :angulo
    avanza (3*:paso) giraderecha :angulo
    avanza (5*:paso) giraderecha :angulo
    avanza (3*:paso)
    haz "angulo cambiasigno :angulo

```

```

        subelapiz
        ponxy :paso (2*:paso)
        bajalapiz
        ponrumbo 0 ]
    fin
c) para cubo
    ponxy 0 0 ponxy 200 0
    ponxy 200 200 ponxy 0 200
    ponxy 0 0 ponxy 0 200
    ponxy 50 250 ponxy 250 250
    ponxy 200 200 ponxy 250 250
    ponxy 250 50 ponxy 200 0
    fin

```

## 24.9. Ejercicios 9.3

1. para raiz\_con\_prueba :numero
 

```

        si (:numero < 0)
          [ escribe [La raiz cuadrada de un negativo no es real ] ]
          [ escribe raizcuadrada :numero ]
        fin

```
2. para no\_menor :n1 :n2
 

```

        si (:n1 < :n2)
          [ escribe (frase :n1 [es menor que] :n2) ]
          [ si (:n1 > :n2)
            [ escribe (frase :n2 [es menor que] :n1)]
            [ escribe (frase :n1 "y :n2 [son iguales]) ] ]
        fin

```
3. para edad\_laboral :edad
 

```

        si y (:edad > 17) (:edad < 65)
          [ escribe [Esta en edad laboral] ]
        fin

```
4. para multiplo :numero :base
 

```

        si 0 = resto :numero :base
          [ escribe (frase :numero [es multiplo de] :base) ]
          [ escribe (frase :numero [no es multiplo de] :base) ]
        fin

```
5. para mismo\_signo :n1 :n2
 

```

        si (producto :n1 :n2) > 0

```

```

    [ escribe [Los numeros tienen el mismo signo] ]
    [ si (producto :n1 :n2) = 0
      [ escribe [Uno de los numeros es cero] ]
      [ escribe [Los numeros tienen distinto signo] ] ]
  fin

```

6. para calificaciones :nota
 

```

        si :nota < 5
          [ escribe [El alumno esta suspenso] ]
        [ si :nota < 6
          [ escribe [El alumno esta aprobado] ]
          [ si :nota < 7
            [ escribe [El alumno tiene un bien] ]
            [ si :nota < 9
              [ escribe [El alumno tiene un notable] ]
              [ escribe [El alumno tiene un sobresaliente] ] ] ] ]
      fin

```
7. para hipotenusa :cateto1 :cateto2
 

```

        haz "hipotenusa (elevacuadrado :cateto1) + (elevacuadrado :cateto2)
        haz "hipotenusa raizcuadrada :hipotenusa
        escribe frase [La hipotenusa de este triangulo rectangulo es] :hipotenusa
      fin
      para elevacuadrado :numero
        devuelve :numero * :numero
      fin

```

## 24.10. Ejercicios 10.3

1. para extraer
 

```

        escribe primero primero [[22 3] [4 5] [8 35]]
      fin

```
2. para prime :listado
 

```

        devuelve elemento 1 :listado
      fin

```
3. para ulti :listado
 

```

        haz "n cuenta :listado
        devuelve elemento :n :listado
      fin

```
4. para triangulo\_vertices
 

```

        leelista [Dame las coordenadas de los vertices de un triangulo] "coordenadas

```

```

haz "x1 primero :coordenadas
haz "y1 elemento 2 :coordenadas
haz "x2 elemento 3 :coordenadas
haz "y2 elemento 4 :coordenadas
haz "x3 elemento 5 :coordenadas
haz "y3 elemento 6 :coordenadas
triangulo :x1 :y1 :x2 :y2 :x3 :y3
fin
para triangulo :x1 :y1 :x2 :y2 :x3 :y3
  subelapiz
  ponposicion lista :x1 :y1
  bajalapiz
  ponposicion lista :x2 :y2
  ponposicion lista :x3 :y3
  ponposicion lista :x1 :y1
fin

```

## 5. para diagonal

```

leelista [Dame el lado de un cuadrado y te dire su diagonal] "lado
haz "diagonal (raizcuadrada 2) * :lado
escribe frase [La diagonal mide] :diagonal
fin

```

## 6. para mengua :listado

```

repite (cuenta :listado)
  [ escribe :listado
    haz "listado menosultimo :listado ]
fin

```

## 7. para inversa :listado

```

haz "inverso [ ]
repite (cuenta :listado)
  [ haz "inverso ponprimero (primero :listado) :inverso
    haz "listado menosprimero :listado ]
escribe :inverso
fin

```

## 8. para maximo

```

leelista [Dame una lista de numeros y te dire el mayor] "listado
haz "maximo primero :listado
repite (cuenta :listado)
  [ haz "compara primero :listado
    si :compara > :maximo
      [ haz "maximo :compara ]

```

```

        haz "listado menosprimero :listado ]
    escribe frase [El maximo de la serie es] :maximo
fin

9. para suprime :n :listado
    haz "nuevo [ ]
    repite (cuenta :listado)
        [ si no (iguales? contador :n)
            [ haz "nuevo ponultimo elemento contador :listado :nuevo ] ]
    escribe :nuevo
fin

10. para adjunta :n :listado1 :listado2
    haz "nuevo [ ]
    repite (cuenta :listado2)
        [ si (iguales? contador :n)
            [ haz "nuevo ponultimo :listado1 :nuevo
                haz "nuevo ponultimo elemento contador :listado2 :nuevo ]
            haz "nuevo ponultimo elemento contador :listado2 :nuevo ]
    escribe :nuevo
fin

```

## 24.11. Ejercicios 11.5.2

1. para potencias :base :exponente
 

```

                si :exponente = 0
                    [ devuelve 1 ]
                [ devuelve :base * potencias :base (:exponente - 1) ]
            fin
            
```
2. para fibonacci :n [:elemento1 1] [:elemento2 1]
 

```

                si :n = 3
                    [ devuelve (:elemento1 + :elemento2) ]
                [ devuelve (fibonacci (:n-1) :elemento2 (:elemento1 + :elemento2)) ]
            fin
            
```
3. para sumapares :n
 

```

                si :n = 0
                    [ devuelve 0 ]
                [ devuelve 2*:n + sumapares (:n-1) ]
            fin
            
```
4. para cuadrados :lado
 

```

                si :lado > 75
            
```



```
    [ alto ]
    [ cuadrado :lado
      cuadrados :lado + 10 ]
  fin
```

```
5. para cuadrados1000 [:numero 1]
    haz "prueba :numero * :numero
    si :prueba > 1000
      [ alto ]
      [ escribe :prueba
        (cuadrados1000 (:numero + 1)) ]
  fin
```