

Capítulo 4

Iniciación a la Geometría de la tortuga

Comenzamos la descripción de las *primitivas* de xLOGO.

4.1. Descripción de las primitivas o comandos

Las primitivas son órdenes básicas que el programa ya tiene incorporadas; al escribirlas en la *línea de comandos* ordenan a la tortuga realizar una acción determinada. La ventaja de LOGO frente a otros lenguajes radica en que las órdenes pueden escribirse en el idioma *natural* del usuario/programador, de modo que resultan muy fáciles de entender.

Cada primitiva puede tener un cierto número de parámetros que son llamados *argumentos*. Por ejemplo, la primitiva `pi`, que devuelve el valor del número π (3.141592653589793), no lleva argumentos: la primitiva `escribe` espera uno (`escribe` muestra en el Histórico de Comandos ese argumento), mientras que la primitiva `suma` (sección 7.1) tiene dos argumentos.

```
escribe suma 2 3 devuelve 5.
```

Los argumentos en LOGO son de tres tipos: **Número**, **Palabra** o **Lista**

- **Números:** Algunas primitivas esperan números como su argumento: `avanza 100` (sección 4.2) es un ejemplo.
- **Palabras:** Las palabras se escriben precedidas por ". Un ejemplo de una primitiva que admite una palabra como argumento es `escribe`.

```
escribe "hola devuelve hola
```

Nota que si olvidas las comillas ("), el intérprete devuelve un mensaje de error. En efecto, `escribe` esperaba ese argumento, pero para el intérprete, `hola` no representa nada, ya que no fue definido como número, ni palabra, ni lista, ni procedimiento.

- **Listas:** Se definen encerrándolas entre corchetes.

escribe [El gato es gris] devuelve El gato es gris

Los números son tratados a veces como un valor (por ej: `avanza 100`), o bien como una palabra (por ejemplo: `escribe vacio? 12` devuelve `falso` – sección 9.4).

Algunas primitivas tienen una forma general, esto es, pueden ser utilizadas con números o *argumentos opcionales*. Estas primitivas son:

- `escribe`
- `suma`, `producto` (sección 7.1)
- `o`, `y` (sección 9.2)
- `lista`, `frase`, `palabra` (sección 10.1)

Para que el intérprete las considere en su forma general, tenemos que escribir las órdenes entre paréntesis. Observa los ejemplos:

`escribe (suma 1 2 3 4 5)`

devuelve:

15

También:

`escribe (lista [a b] 1 [c d])`

devuelve:

[a b] 1 [c d]

y

si (y 1=1 2=2 8=5+3) [avanza 100 giraderecha 90]

4.2. Movimientos de la tortuga

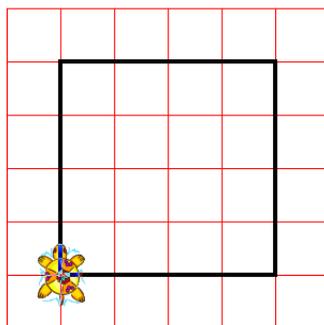
Empecemos por las primitivas que controlan el movimiento de la tortuga:

Primitiva	Forma larga	Forma corta
AVanzar n pasos	avanza n	av n
REtroceder n pasos	retrocede n	re n
Gira Derecha n grados	giraderecha n	gd n
Gira Izquierda n grados	giraizquierda n	gi n
Llevar la tortuga al centro de la pantalla	centro	

Observa que no todas las primitivas son necesarias. Por ejemplo:

- `giraizquierda 90` equivale a `giraderecha -90`
- `retrocede 100` equivale a `avanza -100`

Por ejemplo, para dibujar un cuadrado,



podemos teclear:

```
avanza 200 giraderecha 90 avanza 200 giraderecha 90 av 200 gd 90 av 200
```

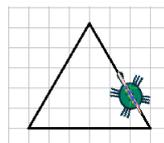
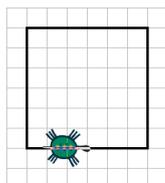
donde puedes ver que podemos utilizar indistintamente las primitivas en su forma larga o en la forma abreviada sin problemas.

	<p>Haz todas las pruebas que necesites para entender perfectamente estas primitivas. Comprueba que la predicción del tema anterior de cómo realiza los giros es correcta, y si has entendido bien el punto de vista de la tortuga</p>
---	--

4.3. Ejercicios

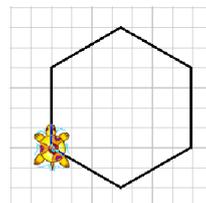
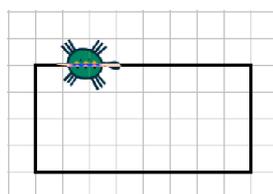
En los dibujos, el lado de cada cuadrado de la cuadrícula mide 25 “pasos de tortuga”.

1. Dibuja el borde de un cuadrado en sentido *antihorario*
2. Dibuja el borde de un cuadrado en sentido *horario*



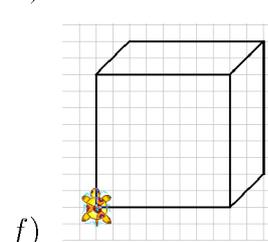
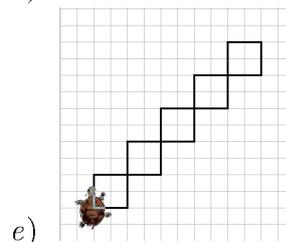
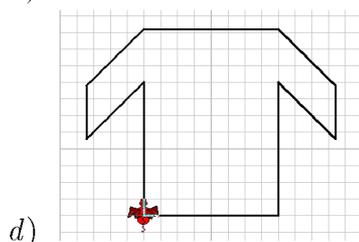
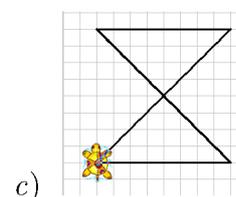
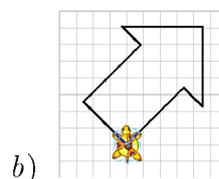
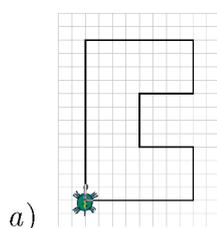
3. Dibuja el borde de un triángulo equilátero en sentido *antihorario*

4. Dibuja el borde de un rectángulo



5. Dibuja el borde de un hexágono regular

6. Dibuja:



	<p>La tortuga tiene muy “mala memoria”, así que todas las órdenes que has dado para dibujar los ejemplos se perderán. Puedes hacer dos cosas: copiarlas en un papel, o usar la Opción del Menú: Archivo → Zona de Texto → Guardar en formato RTF.</p>
---	--

4.4. Avanzando un poco

Continuemos con primitivas que controlan otros aspectos:

Primitiva	Forma larga	Forma corta
Borrar Pantalla y tortuga al centro	<code>borrarapantalla</code>	<code>bp</code>
Subir Lápiz (no deja trazo al moverse)	<code>subelapiz</code>	<code>sl</code>
Bajar Lápiz (sí deja trazo al moverse)	<code>bajalapiz</code>	<code>bl</code>
Ocultar Tortuga	<code>ocultatortuga</code>	<code>ot</code>
Mostrar Tortuga	<code>muestratortuga</code>	<code>mt</code>

Primitiva	Forma larga	Forma corta
Cambiar el color del trazo con que dibuja	poncolorlapiz n	poncl n
Cambiar el grosor del trazo con que dibuja	pongrosor n	
Borrar por donde pasa, en vez de escribir	goma	go
Dejar de borrar y volver a escribir	bajalapiz	bl
Rellenar con el color activo una región cerrada	rellena	
Rellenar una región limitada por el color activo	rellenazona	
Limpia la pantalla dejando la tortuga en el sitio	limpia	
Repetir n veces lo indicado entre corchetes	repite n	

La primitiva `poncolorlapiz` debe ir acompañada de un número (ver sección 13.2.2), y las opciones con las que trabajaremos de momento serán:

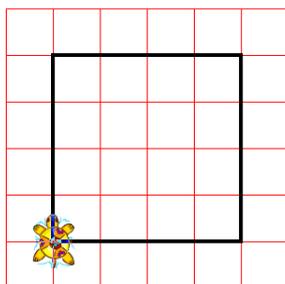
0: negro 1: rojo 2: verde 3: amarillo
4: azul 5 magenta 6: cyan 7 blanco

Respecto a `repite`, las órdenes a repetir deben ir entre corchetes, por ejemplo:

```
repite 4 [escribe "Hola ]
```

escribe 4 veces la palabra Hola

Por ejemplo, para dibujar el mismo cuadrado del ejemplo anterior:



habíamos escrito:

```
av 200 gd 90 av 200 gd 90 av 200 gd 90 av 200
```

Es fácil observar que hay órdenes que se repiten cuatro veces. Si pensamos un momento, podemos ver que añadir un `giraderecha` más no va a modificar el dibujo, así que podremos escribir:

```
repite 4 [ avanza 200 giraderecha 90 ]
```

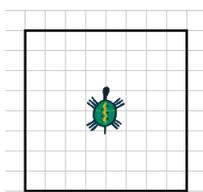
con el mismo resultado.



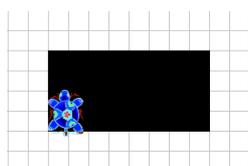
Analiza los programas con los que dibujaste antes los cuadrados, el triángulo y el hexágono. ¿Ves cómo se repiten varias veces las mismas órdenes? Aplica lo que acabamos de ver con la primitiva `repite` para hacer más sencillos tus programas.

4.5. Ejercicios

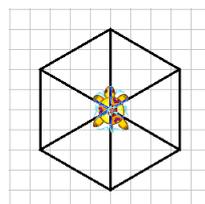
1. Dibuja el borde de un cuadrado, pero ahora usa la primitiva `repite`
2. Dibuja el borde de un triángulo equilátero usando la primitiva `repite`
3. Dibuja el borde de un hexágono regular usando la primitiva `repite`
4. Dibuja el borde de un cuadrado, cuyo centro esté en el centro de la pantalla
5. Dibuja un rectángulo, rellenando el interior



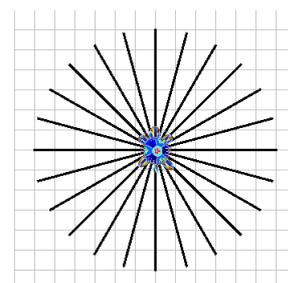
Problema 4



Problema 5



Problema 6



Problema 7

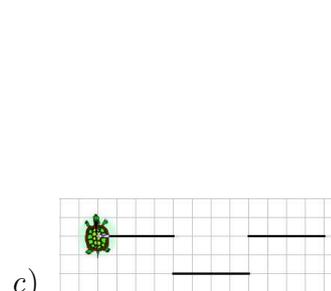
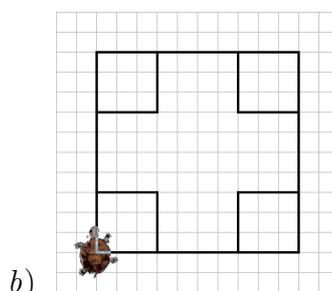
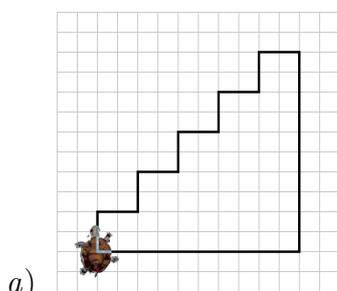
De nuevo, el lado de cada cuadrado de la cuadrícula mide 25 “pasos de tortuga”

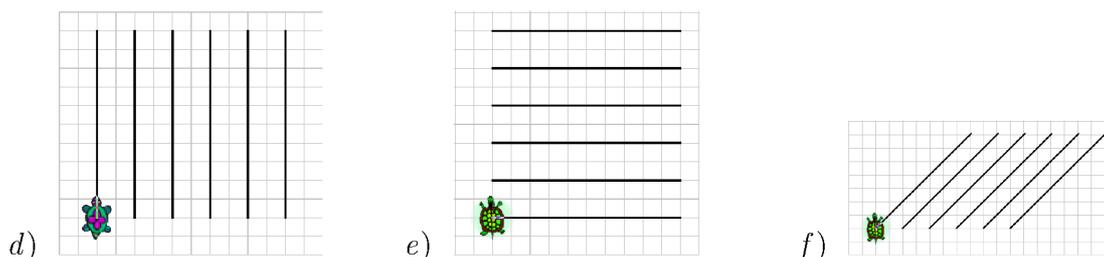
6. Dibuja el borde de un hexágono regular y las diagonales cuyos extremos son dos vértices opuestos del mismo

	<p>Acabas de dibujar polígonos de 3, 4 y 6 lados. ¿Serías capaz de determinar una regla para hallar el ángulo de giro en cualquier polígono?</p> <p>Observa que la tortuga da una vuelta completa alrededor del polígono, que al final del proceso vuelve a estar mirando hacia arriba y fíjate cuántas veces tiene que girar.</p>
--	---

7. Dibuja los radios de una rueda. En total tienen que salirte 24

8. Dibuja:





	<p>Recuerda copiar las respuestas a los problemas en un papel, o guardarlas en el disco duro: Archivo → Zona de Texto → Guardar en formato RTF.</p>
---	--

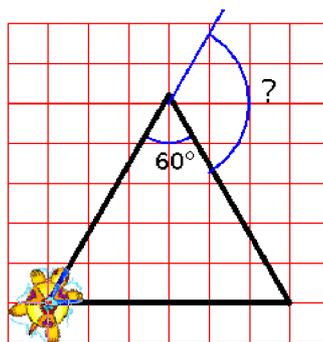
	<p>¿Qué te parece este método para dibujar? ¿Crees que puede mejorarse? ¿Cómo? ¿Qué pasa si quiero que los polígonos tengan lados más largos o más cortos? ¿Y si quiero cambiar el tamaño de las figuras que acabas de conseguir?</p>
---	---

4.6. Aplicación didáctica de xLogo

Ya hemos visto cómo dibujar el cuadrado, pero podemos analizar un poco mejor el proceso que hemos seguido para hacerlo.

4.6.1. El triángulo equilátero

Vamos a ver cómo trazar este triángulo equilátero de 180 pasos de tortuga:



Aquí, un cuadrado representa 30 pasos de tortuga

Las órdenes serán algo del estilo:

```
repite 3
[ avanza 180 giraderecha ... ]
```

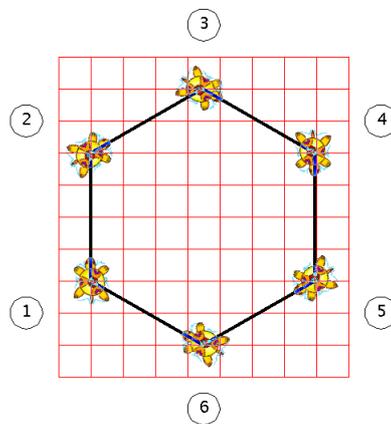
Queda por determinar el ángulo correcto. En un triángulo equilátero, los ángulos valen todos 60 grados, y como la tortuga debe volver por el exterior del triángulo, el ángulo valdrá:

$$180 - 60 = 120 \text{ grados}$$

Las órdenes son, pues:

```
repite 3
  [ avanza 180 giraderecha 120 ]
```

4.6.2. El hexágono



Un cuadrado = 20 pasos de tortuga.

Para un cuadrado repetíamos 4 veces, para el triángulo 3 veces, parece claro que para el hexágono será:

```
repite 6
  [ avanza 80 giraderecha ... ]
```

Date cuenta que en su desplazamiento, la tortuga realmente da una vuelta completa sobre ella misma. (Inicialmente está orientada hacia arriba y termina en esta misma posición). Esta rotación de 360 grados se efectúa en 6 etapas. Por tanto, cada vez, gira

$$360/6 = 60 \text{ grados}$$

Las órdenes son, entonces:

```
repite 6
  [ avanza 80 giraderecha 60 ]
```

4.6.3. Trazar un polígono regular en general

En realidad, reiterando el razonamiento anterior, puedes darte cuenta de que para trazar un polígono de n lados, el ángulo se obtendrá dividiendo 360 por n . Por ejemplo:

- Para trazar un pentágono regular de lado 100: $(360:5=72)$

```
repite 5 [ avanza 100 giraderecha 72 ]
```

- Para trazar un eneágono regular de lado 20: $(360:9=40)$

```
repite 9 [ avanza 20 giraderecha 40 ]
```

- Para trazar un eh ... 360-gono¹ regular de lado 2 (que se parece mucho a un círculo):

```
repite 360 [ avanza 2 giraderecha 1 ]
```

- Para trazar un heptágono regular de lado 120:

```
repite 7 [ avanza 120 giraderecha 360/7 ]
```

En el siguiente capítulo vamos a aprender cómo evitar que *se pierdan* las órdenes que hemos dado a la tortuga, de un modo más simple que guardando en RTF y copiando.

4.7. Función avanzada de relleno

xLOGO posee una tercera primitiva de relleno, `rellenapoligono`. Su único argumento es una lista que debe contener las instrucciones para dibujar una figura poligonal cerrada:

```
rellenapoligono [ lista_de_instrucciones]
```

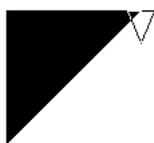
Esta primitiva rellena la forma creada por triangulación (utilizando una serie de triángulos), de modo que cada vez que la tortuga dibuja una línea, el triángulo que se genera es relleno con el color activo.

Por ejemplo:

```
rellenapoligono [ repite 4 [avanza 100 giraderecha 90]]
```



Paso 1



Paso 2



Paso 3



Paso 4

¹Trihectahexacontágono: tri = 3, hecta = 100, hexaconta = 60, gono = ángulo

Utilizado adecuadamente, puede proporcionar resultados llamativos:

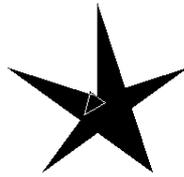
```
repite 5  
  [ avanza 100 rellenapoligono [ retrocede 100 giraderecha 144 avanza 100 ]  
    avanza 100 giraizquierda 72]
```



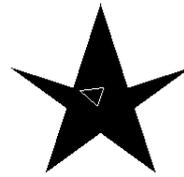
Paso 1



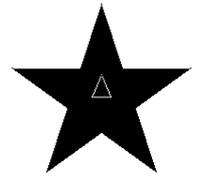
Paso 2



Paso 3



Paso 4



Paso 5