

## Capítulo 8

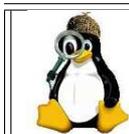
# Coordenadas y Rumbo

¿Cómo podemos saber dónde se encuentra la tortuga? ¿Puedo enviarla a un punto concreto de la pantalla con una sola orden? ¿Hacia dónde está *mirando*? Estas preguntas se responden en este capítulo.

### 8.1. Cuadrícula y ejes

Comencemos por las primitivas que pueden ayudarnos enormemente en nuestro trabajo con las coordenadas:

Primitivas	Argumentos	Uso
cuadrícula	a b (números)	Dibuja una cuadrícula en el <b>Área de dibujo</b> de dimensiones a x b y borra la pantalla
detienecuadrícula	no	Quita la cuadrícula del <b>Área de dibujo</b> y borra la pantalla
poncolorcuadrícula, pcc	primitiva, lista o numero	Establece el color de la cuadrícula del <b>Área de dibujo</b>
ejes	número	Dibuja los ejes cartesianos (X e Y) de escala (separación entre marcas) n, con las etiquetas correspondientes.
ejex	número	Dibuja el eje de abscisas (eje X) de escala (separación entre marcas) n, con las etiquetas correspondientes.
ejey	número	Dibuja el eje de ordenadas (eje Y) de escala (separación entre marcas) n, con las etiquetas correspondientes.
detieneejes	ninguno	Quita los ejes del <b>Área de dibujo</b> y borra la pantalla
poncolorejes, pce	primitiva, lista o numero	Establece el color de los ejes en el <b>Área de dibujo</b>



**El sistema cartesiano con coordenadas ortogonales no es el único que existe.** Investiga otros sistemas de representación e intenta crear procedimientos que los muestren en pantalla.

**Nota:** `detienecuadrícula` y `detieneejjes` retiran la cuadrícula y los ejes de pantalla, y también la borra del mismo modo que `borrapantalla`

## 8.2. Coordenadas

El **Área de Dibujo** es un sistema cartesiano cuyo origen está situado en el centro de la pantalla. De este modo, podemos alcanzar cualquiera de los puntos de dicho área ayudados por sus coordenadas. Las primitivas asociadas son:

Primitiva	Forma larga	Forma corta
Mostrar la posición (devuelve una lista)	<code>posicion</code>	<code>pos</code>
Mostrar sólo la abscisa (coordenada X)	<code>primero posicion coordenadax</code>	<code>pr pos coordx</code>
Mostrar sólo la ordenada (coordenada Y)	<code>ultimo posicion coordenaday</code>	<code>ultimo pos coordy</code>
Mover al punto [X,Y] (X,Y números)	<code>ponposicion [X Y]</code>	<code>ponpos [X Y]</code>
Pintar el punto de coordenadas [X,Y]	<code>punto [X Y]</code>	
Mover hacia [X,Y] (X,Y números o variables)		<code>ponxy :X :Y</code>
Llevar hacia el punto de abscisa X (número o variable)		<code>ponx :X</code>
Llevar hacia el punto de ordenada Y (número o variable)		<code>pony :Y</code>

De nuevo debemos tener cuidado con la prioridad de las primitivas. Si alguna coordenada es negativa, debemos usar paréntesis:

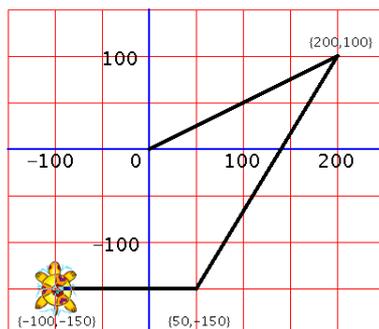
```
ponxy 100 (-60)
```

Para usar `ponposicion` con variables, debemos usar la primitiva frase:

```
ponposicion frase :abscisa :ordenada
```

Un ejemplo sencillo de su utilización:

```
borrapantalla
ponpos [200 100]
ponpos [50 -150]
ponpos [-100 -150]
```



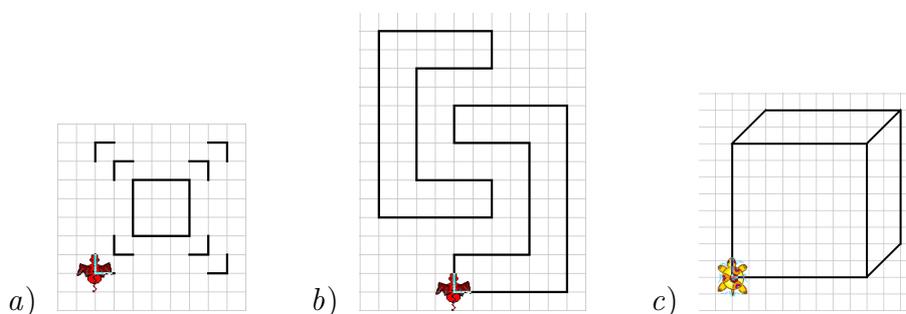
### 8.3. Ejercicios

1. Dibuja un rectángulo usando solamente las primitivas `ponx` y `pony`
2. Dibuja un triángulo rectángulo isósceles usando únicamente `ponposicion`
3. Construye el procedimiento `segmento`, cuyas 4 entradas sean las coordenadas de dos puntos y que dibuje el segmento cuyos extremos son esos dos puntos.
4. Define el procedimiento `cuadrilatero`, cuyas OCHO entradas sean las coordenadas de cuatro puntos y que dibuje el cuadrilátero cuyos vértices son esos ocho puntos.
5. Plantea un procedimiento `dist_ptos`, que a partir del procedimiento `segmento` calcule la distancia entre los dos puntos (o lo que es lo mismo, la longitud del segmento)

**Dato:** La distancia entre dos puntos  $(x_0, y_0)$  y  $(x_1, y_1)$  se calcula mediante la fórmula:

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

6. Define el procedimiento `triangulo`, cuyas SEIS entradas sean los vértices de un triángulo, lo dibuje y lo rellene.
7. Dibuja:

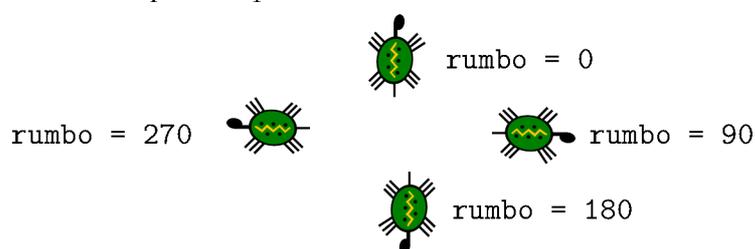


## 8.4. Rumbo

Las primitivas asociadas al rumbo son:

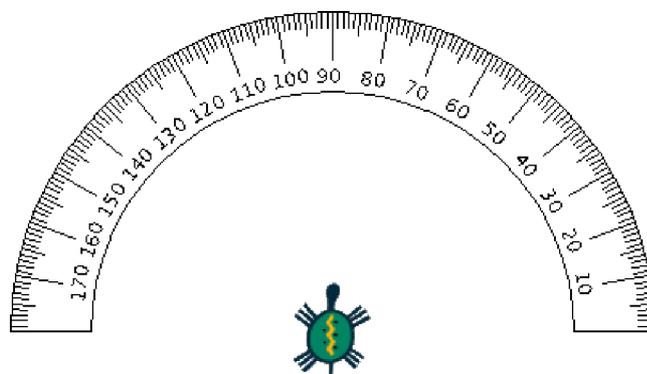
Primitiva	Forma larga	Forma corta
Orientar la tortuga hacia un rumbo $n$	<code>ponrumbo :n</code>	<code>ponr :n</code>
Pedir el rumbo (en grados respecto de la vertical y sentido horario)		<code>rumbo</code>
Para volver al origen con rumbo 0 (mirando hacia arriba)		<code>centro</code>
Pedir el rumbo que la tortuga debe seguir hacia el punto $[X Y]$		<code>hacia [X Y]</code>
Pedir la distancia (en pasos) hasta el punto $[X Y]$		<code>distancia [X Y]</code>

Una imagen vale más que mil palabras:



## 8.5. Ejercicios

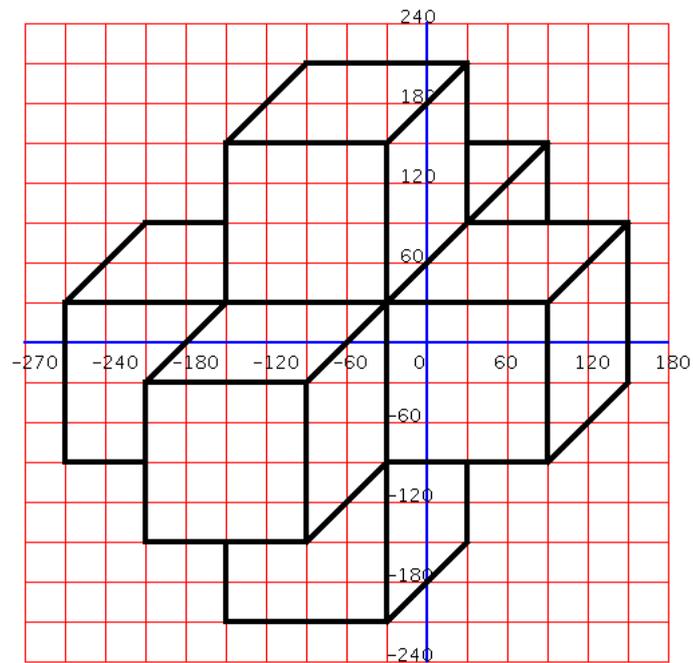
1. Define procedimientos que pongan a la tortuga rumbo a los puntos cardinales: **norte**, **sur**, **este**, **oeste**, **noroeste**, **nordeste**, **sudeste** y **suroeste**.
2. En la sección anterior te pedíamos que buscaras otros sistemas de coordenadas. Te pedimos aquí que crees un procedimiento que represente en pantalla un sistema de coordenadas **polares**
3. Para manejar bien los rumbos es conveniente tener a la vista un transportador. Crea un procedimiento **transportador** que dibuje uno en pantalla:



Para indicar la escala, puedes usar la primitiva **rotula** (sección 12.1)

## 8.6. Actividad avanzada

En esta actividad debes realizar la siguiente figura.



Sólo puedes utilizar las primitivas

- `ponposicion` (`ponpos`)
- `borrarantalla` (`bp`)
- `subelapiz` (`sl`)
- `bajalapiz` (`bl`)