

## TUTORIAL DE TORTUGARTE

### Introducción

TortugArte es una actividad inspirada en LOGO.

Es un lenguaje de programación. Sin embargo, y más importante, es un lenguaje para aprender. Es una herramienta útil para enseñar el proceso de aprendizaje y de pensamiento.

Consiste en programar el comportamiento de una tortuga, realizando formas y diseños en la pantalla de la computadora.

Es fácil de usar y muy intuitivo, existe una amplia variedad de bloques que modelan el comportamiento de la tortuga.

Existen diferentes paletas que contienen un grupo de bloques relacionados entre si. Hasta el momento las paletas son:

- Tortuga: Describe los movimientos de la tortuga, sus coordenadas y su rumbo.
- Pluma: Diseño del rastro de la tortuga, tamaño y color de la pluma.
- Números: Permite expresiones algebraicas, operaciones básicas y comparaciones entre valores.
- Sensores: Describe diferentes valores en los sensores de la XO.
- Flujo: Contiene bloques de flujo, condiciones y repeticiones.
- Mis bloques: Bloques definidos por el usuario.

### Empezando a utilizar el TortugArte

Darle movimiento a la tortuga es muy simple, sólo hay que indicarle que vaya hacia adelante con el bloque “adelante”. Si le damos el valor **adelante 100** le estamos diciendo a la tortuga que avance 100 píxeles. El *pixel* es la unidad de medida de la resolución de la pantalla, en la XO es de 1200x900 píxeles y el TortugArte usa 1200x800 px.

Además de hacer andar hacia adelante la tortuga se le puede decir que gire hacia la derecha o la izquierda. Si le damos la instrucción **derecha 90** le estamos diciendo que gire hacia su derecha un ángulo de 90 grados.

Con esas dos instrucciones podemos hacer nuestro primer ejemplo, un cuadrado:

```
adelante 100  
derecha 90  
adelante 100  
derecha 90  
adelante 100  
derecha 90  
adelante 100  
derecha 90
```

Como se puede ver se ejecuta varias veces el mismo patrón (adelante 100; derecha 90). Para evitar la repetición de código existen los bloques de repetición, el ejemplo anterior quedaría de la siguiente manera:

```
repetir 4  
  adelante 100  
  derecha 90
```

El código que se encuentre dentro del bloque “**repetir**” se va a ejecutar tantas veces como se le indique (en este caso 4). Con este tipo de bloques acortamos el código y resulta más fácil de entender, lo que antes llevaba 8 líneas ahora lleva 3.

También se pueden definir bloques de usuario con los bloques llamados “**pila**”, esto sirve para “hacer más prolijo” el código y para no repetirlo en el caso que se necesite usar las mismas instrucciones en más de una parte del código.

El ejemplo del cuadrado quedaría:

```
pila1{
  repetir 4
    adelante 100
    derecha 90
}
```

Ahora cada vez que se ejecute la instrucción “**pila1**” se estará ejecutando todo el código que contiene.

Podemos dibujar el cuadrado rotado escribiendo el siguiente código:

```
limpiar
izquierda 36
pila1
```

*Pila1* es el bloque que definimos previamente.

Aquí aparece un bloque nuevo llamado “**limpiar**”, lo que hace ese bloque es limpiar la pantalla, borra todo lo que había anteriormente y coloca la tortuga en el centro de la pantalla.

Si seguimos haciendo crecer el código y combinamos lo que se ha hecho hasta ahora...

```
limpiar
repetir 10
  pila1
  izquierda 36
```

con este código hacemos una especie de “estrella” formada por cuadrados rotados.

Ahora agregaremos los bloques de usuario llamados “**caja**”. La función de esos bloques es llevar variables del usuario. Por ejemplo si en un ciclo de repetición quieren ir incrementando un valor de una instrucción.

Ejemplo:

```
limpiar
poner_en_caja1 1
repetir 10
  pila1
  izquierda 36
  fijar_color caja1
  poner_en_ (caja1 caja1 + 9)
```

En este ejemplo además del concepto “caja” se incorporan dos bloques más. El bloque “**fijar\_color**” que fija el color para el pincel, el rango de colores va de 0 a 100. Así como el bloque “**fijar\_color**” existe el bloque “**fijar\_tono**” que fija la tonalidad del color del pincel. El otro bloque que se muestra es el bloque “+” que es la suma numérica, en este caso sirve para cambiar el color del pincel.

Haciendo unos retoques a los ejemplos anteriores llegamos al ejemplo final:

```
pila1 {  
  repetir 4  
    adelante caja1  
    derecha 90  
}  
  
pila2 {  
  repetir 10  
    pila1  
    izquierda 36  
}
```

*Programa principal.*

```
limpiar  
poner_en_caja1 0  
repetir 250  
  pila2  
  fijar_color caja1  
  poner_en_caja1 (caja1 + 1)  
  fijar_tono caja1
```

**GUIA DE TORTUGARTE**

**Paleta “Tortuga”**

<b>Bloque</b>	<b>Descripción</b>
<i>limpiar</i>	Borra la pantalla y deja la tortuga en el centro de la misma
<i>adelante</i>	Mueve la tortuga tantos pixeles como se le indique en el sentido que esta trae.
<i>atrás</i>	Mueve la tortuga tantos pixeles como se le indique en el sentido opuesto al que trae.
<i>izquierda</i>	Gira la tortuga a la izquierda tantos grados como se le indique.
<i>derecha</i>	Gira la tortuga a la derecha tantos grados como se le indique.
<i>arco</i>	Realiza un arco de ángulo y radio indicados
<i>fijar xy</i>	Fija la posición de la tortuga en las coordenadas (x,y) indicadas siendo la posición (0,0) el centro de la pantalla y van desde (-600,-400) hasta (600,400)
<i>rumbo</i>	Fija el rumbo de la tortuga, es un valor entre 0 y 360
<i>coorx</i>	Retorna el valor de la coordenada x
<i>coory</i>	Retorna el valor de la coordenada y
<i>rumbo</i>	Retorna el valor del rumbo de la tortuga

**Paleta “Pincel”**

<b>Bloque</b>	<b>Descripción</b>
<i>cp</i>	Levanta el pincel
<i>sp</i>	Baja el pincel
<i>fijar tamaño</i>	Fija el grosor del pincel
<i>fijar color</i>	Fija el color del pincel
<i>fijar tono</i>	Fija el tono del color del pincel
<i>pintar fondo</i>	Pinta el fondo con el color y el tono especificados
<i>tamaño</i>	Retorna el tamaño del pincel
<i>color</i>	Retorna el color del pincel
<i>tono</i>	Retorna el tono del pincel

<b>Paleta “Números”</b>	
<b>Bloque</b>	<b>Descripción</b>
<i>número</i>	Para asignar valores numéricos a determinado bloque.
<i>+, - , x , /</i>	Realiza las operaciones suma, resta, multiplicación y división.
<i>mod</i>	Retorna el resto de la división entera entre dos números.
<i>azar</i>	Retorna un número al azar dentro del rango especificado.
<i>&lt;, &gt;, =</i>	Compara dos valores y devuelve Verdadero o Falso.
<i>y</i>	Compara 2 expresiones booleanas y devuelve Verdadero si ambas expresiones son Verdaderas.
<i>o</i>	Compara 2 expresiones booleanas y devuelve Verdadero si alguna de las expresiones es Verdadera.
<i>no</i>	Cambia el valor de la expresión booleana de Verdadero a Falso y viceversa.
<i>mostrar</i>	Muestra el valor de la variable.
<b>Paleta “Sensores”</b>	
<b>Bloque</b>	<b>Descripción</b>
<i>nivel de sonido</i>	Retorna el nivel de sonido de la XO.
<i>tono</i>	Retorna el tono de la XO
<i>resistencia</i>	Retorna la resistencia de la XO
<i>voltaje</i>	Retorna el voltaje de la XO
<b>Paleta “Flujo”</b>	
<i>esperar</i>	Detiene la tortuga el tiempo que se le indique.
<i>repetir</i>	Repite las instrucciones contenidas tantas veces como se le indique.
<i>por siempre</i>	Repite las instrucciones indicadas para siempre.
<i>si entonces</i>	Ejecuta el código contenido si se cumple la condición.
<i>si entonces, si no</i>	Si se cumple la condición ejecuta el código que está en “si”, de lo contrario ejecuta el bloque “si no”.
<i>detener pila</i>	Detiene la ejecución de un ciclo.
<b>Colores</b>	

Puede encontrar la referencia de los colores en:  
[http://wiki.laptop.org/go/Image:Turtle\\_art\\_colors.jpg](http://wiki.laptop.org/go/Image:Turtle_art_colors.jpg)