

GUÍA DIDÁCTICA OPERADORES



Preconocimiento

Deberías saber moverte con cierta soltura sobre el entorno y conocer los bloques de movimiento, apariencia, sonido, lápiz, control y sensores. Tu experiencia con Scratch es mucho mayor y deberías conocer muchas de las posibilidades del entorno.



Objectivos

- Conocer los bloques de la familia de operadores.
 - Realizar actividades básicas para comprobar su uso.
-



Temporalización

Contenidos	Temporalización
Familia de bloques de operadores	2 horas

Contenidos



Indice

1. Introducción
 2. Un poco de teoría
 1. Cuestión de formas
 2. Descripción del funcionamiento de los bloques
 3. Los bloques uno a uno
 4. Cambiando de operador
 5. Anidando cálculos
 3. Actividad guiada
 4. Bibliografía
-

1. Introducción



OPERADORES



En la categoría OPERADORES encontramos todos aquellos bloques que no tienen funciones específicas sobre el entorno pero permiten realizar cálculos y comparaciones que junto a los bloques de otras familias le dan mucha potencia al entorno de programación Scratch.

Estos bloques nos van a permitir realizar operaciones aritméticas básicas, cálculos con funciones, operaciones de comparación e igualdad, operaciones lógicas y operaciones con cadenas de caracteres.

2. Un poco de teoría



CUESTIÓN DE FORMAS

Os habréis dado cuenta que las formas de los distintos bloques existentes en Scratch no están hechas al azar, en función de una forma u otra Scratch nos está indicando que tipo de bloques están permitidos unir o insertar, existen 3 formas básicas que afectan a los operadores:

Forma	Bloques que podemos insertar	Ejemplos
	Todos aquellos cuyo resultado es un número.	 
	Todos aquellos cuyo resultado es un número o una cadena de caracteres.	 
	Todos aquellos cuyo resultado es lógico (verdadero o falso).	 



DESCRIPCIÓN DE LOS BLOQUES

Tabla con la descripción del funcionamiento de cada uno de los bloques:

OPERADORES



Devuelve la suma de los dos operadores introducidos.



Devuelve la resta de los dos operadores introducidos.



Devuelve la multiplicación de los dos operadores introducidos.



Devuelve la división de los dos operadores introducidos.



Devuelve un número al azar entre



Devuelve verdadero en el caso de que el primer operador sea menor que el segundo.



Devuelve verdadero en el caso de que el primer operador sea igual que el segundo.



Devuelve verdadero en el caso de que el primer operador sea mayor que el segundo.



Devuelve verdadero en el caso de que el primer operador y el segundo sean verdaderos. Resultado de la función "y lógica".



Devuelve verdadero en el caso de que el primer operador o el segundo sean verdaderos. Resultado de la función "o lógica".



Devuelve la negación del operador introducido.



Devuelve una cadena resultante de la unión de dos cadenas.



Devuelve la letra que ocupa la posición indicada en el primer operador.



Devuelve la longitud de la cadena de caracteres introducida.



Devuelve el resto obtenido al calcular la división de los dos operadores.



Devuelve el resultado de redondear a un número entero el operador introducido.



Devuelve el resultado de la función indicada sobre el operador.

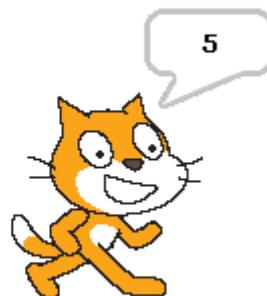
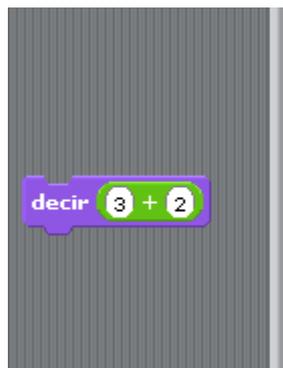


LOS BLOQUES UNO A UNO

EL BLOQUE "SUMA"



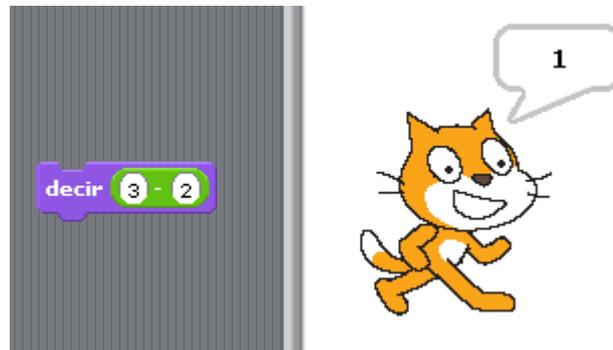
Devuelve la suma de los dos operadores introducidos.



EL BLOQUE "RESTA"



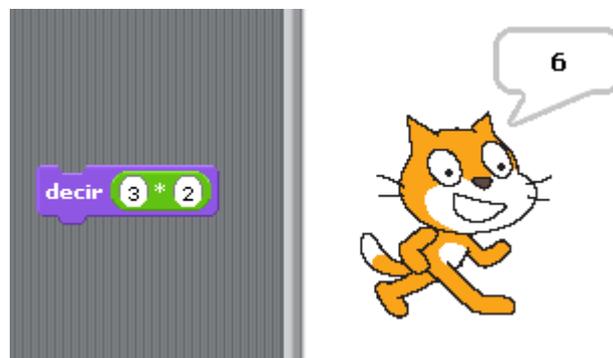
Devuelve la resta de los dos operadores introducidos.



EL BLOQUE "MULTIPLICACIÓN"



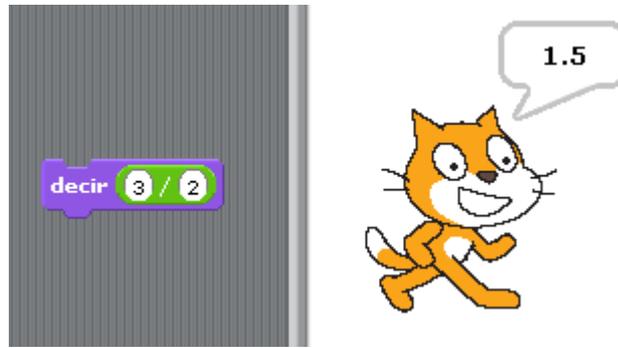
Devuelve la multiplicación de los dos operadores introducidos.



EL BLOQUE "DIVISIÓN"



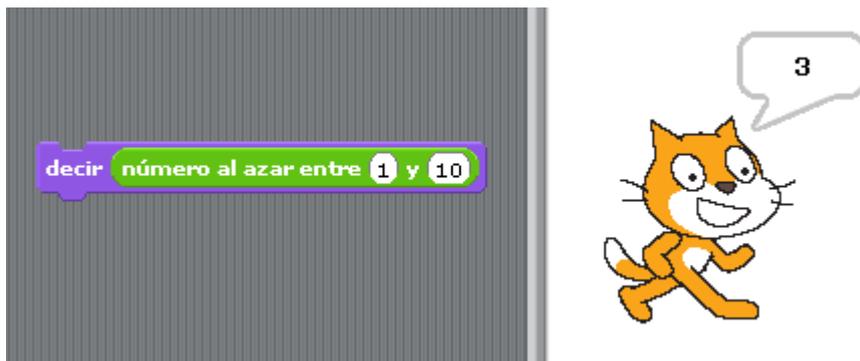
Devuelve la división de los dos operadores introducidos.



EL BLOQUE "NÚMERO AL AZAR"



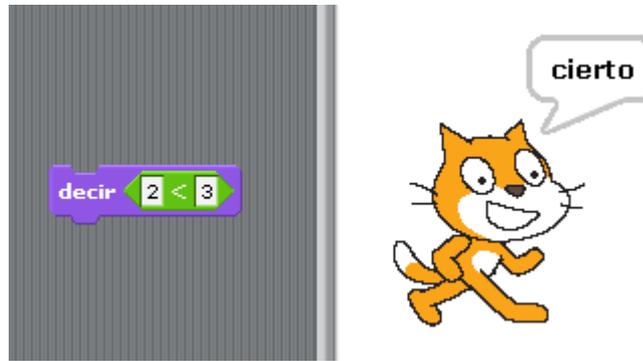
Devuelve un número entero al azar comprendido entre los dos operadores introducidos.



EL BLOQUE "MENOR QUE"



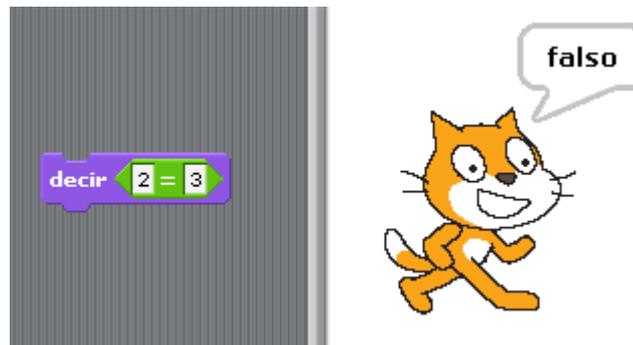
Devuelve verdadero si el primer operador es menor que el segundo, falso en caso contrario.



EL BLOQUE "IGUALDAD"



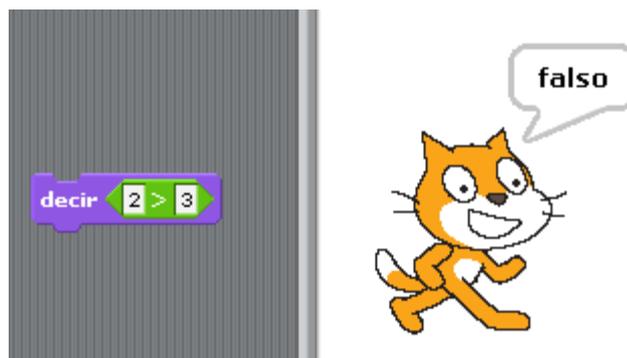
Devuelve verdadero en el caso de que los dos operadores sean iguales, falso si son distintos.



EL BLOQUE "MAYOR QUE"



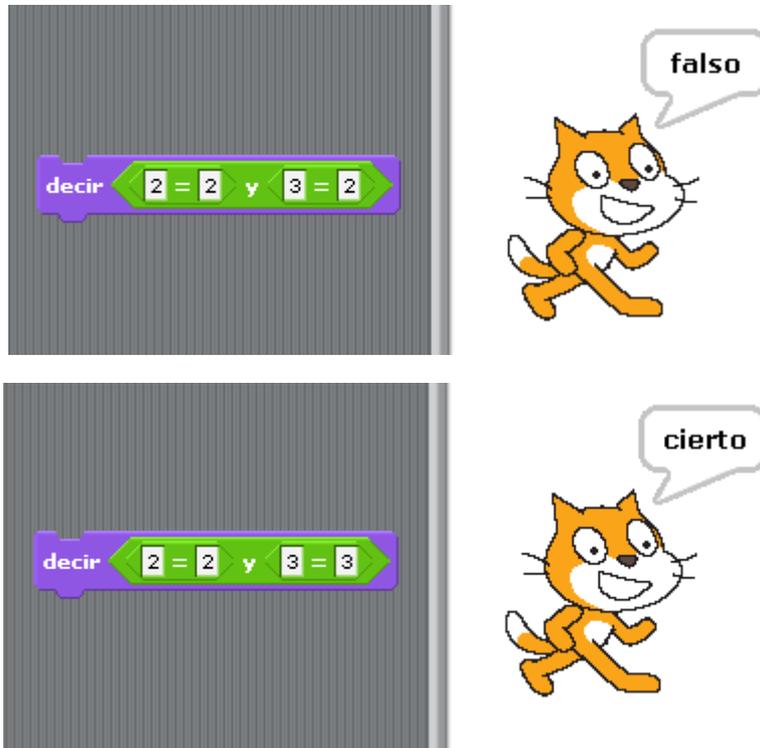
Devuelve verdadero en el caso de que el primer operador sea mayor que el segundo operador, falso en caso contrario.



EL BLOQUE "Y"



Operador lógico que devuelve verdadero en el caso de que los dos operadores sean verdaderos, falso en cualquier otro caso.

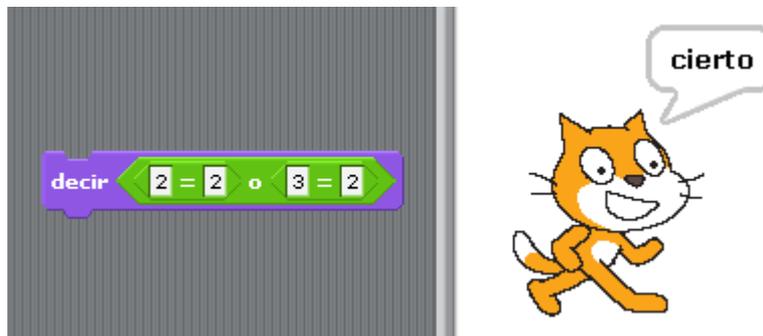


Operador 1	Operador 2	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Falso
Falso	Verdadero	Falso
Falso	Falso	Falso

EL BLOQUE "O"



Operador lógico que devuelve verdadero cuando alguno de los dos operadores es verdadero, falso cuando los dos operadores son falsos.

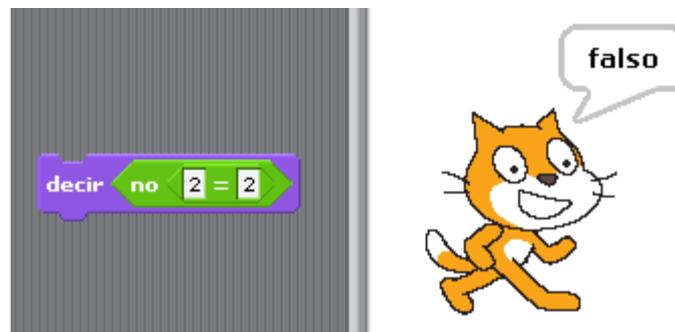


Operador 1	Operador 2	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	Verdadero
Falso	Verdadero	Verdadero
Falso	Falso	Falso

EL BLOQUE "NEGACIÓN"



Operador lógico que devuelve el valor negado del operador, si es verdadero devuelve falso y si es falso devuelve verdadero.

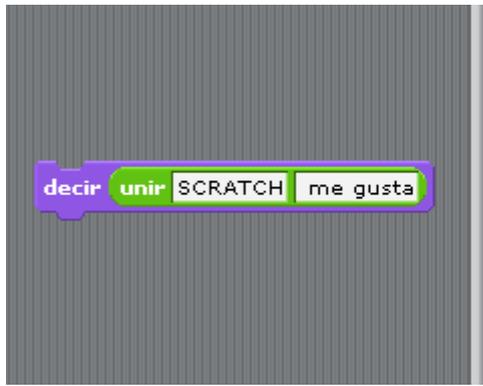


Operador	Resultado
Verdadero	Falso
Falso	Verdadero

EL BLOQUE "CONCATENACIÓN"



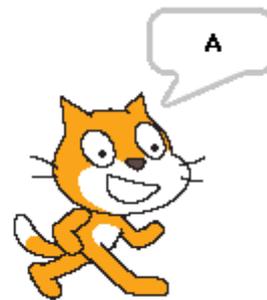
Operador con cadenas de caracteres que devuelve una cadena como resultado de concatenar las dos cadenas pasadas como operadores.



EL BLOQUE "SUBCADENA"



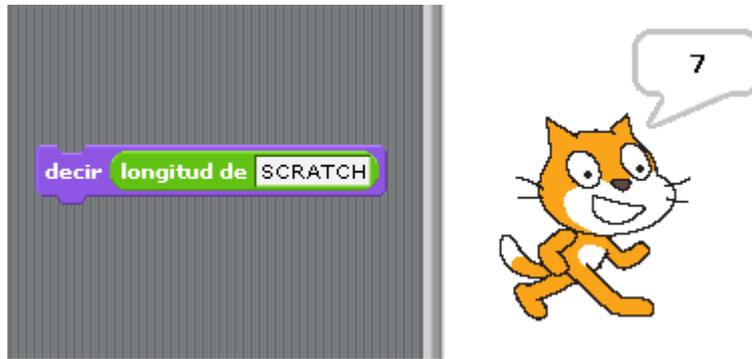
Operador con cadenas de caracteres que devuelve la letra del segundo operador que ocupa la posición indicada por el primer operador.



EL BLOQUE "LONGITUD"



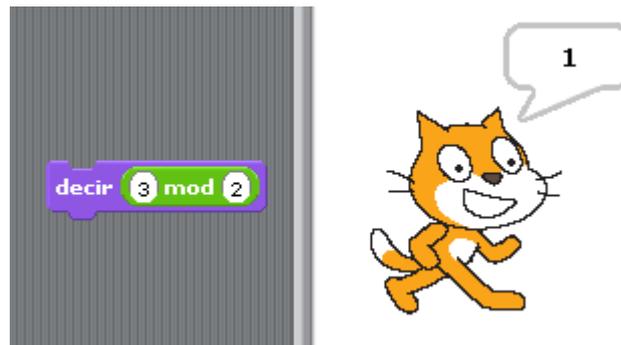
Devuelve el número de letras que contiene la cadena pasada como parámetro.



EL BLOQUE "RESTO"



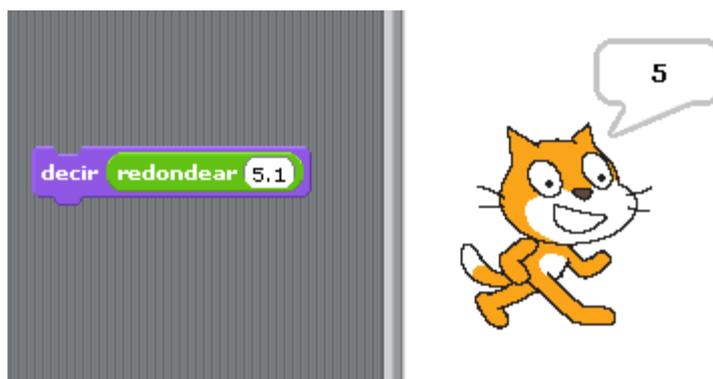
Devuelve el resto resultante de aplicar la división entera entre el primer operador y el segundo.

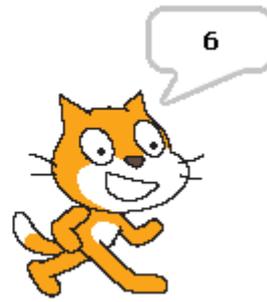


EL BLOQUE "REDONDEAR"



Devuelve el redondeo entero del número pasado como parámetro.



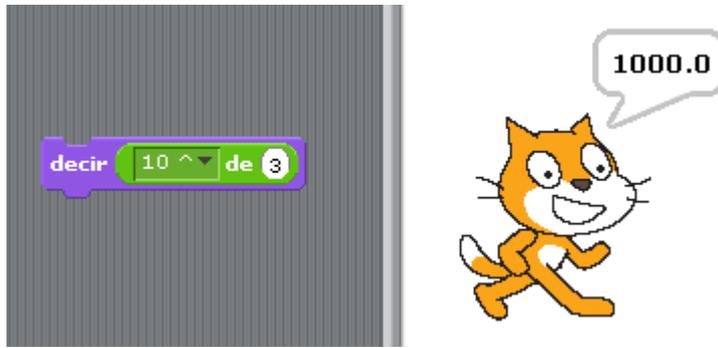


EL BLOQUE "FUNCIÓN"

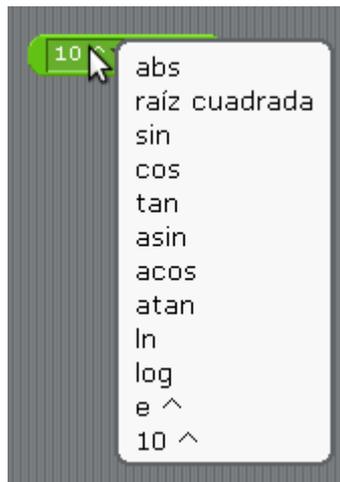


Devuelve el resultado de aplicar la función indicada al número pasado como parámetro.



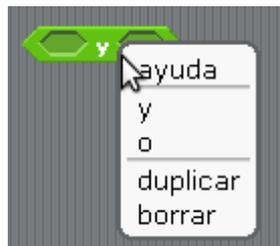
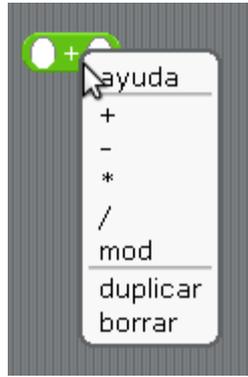


Las funciones que permite elegir son:



CAMBIANDO DE OPERADOR

Cuando por las necesidades de la lógica del programa o simplemente si nos hemos equivocado y necesitamos cambiar el operador de un bloque lógico o aritmético, no siempre es necesario rehacer todos los bloques para cambiar por ejemplo una suma por una resta, o un "y lógico" por un "o lógico". Tenemos un atajo en el menú contextual del bloque, pulsamos el botón derecho y podemos acceder al cambio de operador sin necesidad de ir arrastrando bloques:



Esto puede ayudaros a hacer modificaciones a vuestros programas con más rapidez.



ANIDANDO CÁLCULOS

No siempre los cálculos que podemos realizar con los operadores de Scratch son tan sencillos como los indicados en ejemplos anteriores en que los cálculos sólo afectaban a dos números o cadenas introducidos en el bloque:

$$1 + 1$$

En la realidad puede darse el caso en que los resultados de un bloque se utilicen como un nuevo operador para otro bloque, esto es lo que se conoce como anidar resultados y es habitual en cualquier lenguaje de programación.

Podríamos tener cálculos más complejos como:

$$1 + 1 + 3 + 4$$

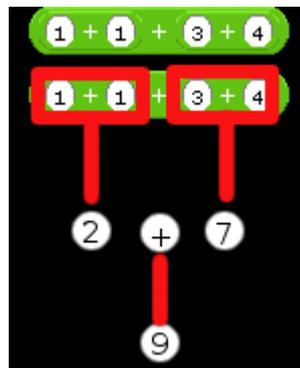
ó

$$\sqrt{10 + 2 * 2 + 5 - 4 + 4}$$

ó

$$x \text{ del ratón} < 12 \text{ y tamaño} = 100$$

En cualquier caso la resolución del resultado que se obtiene utilizando este esquema siempre es igual. Utilicemos el siguiente ejemplo:



Está claro que el resultado será 9, pero ¿cómo hay que realizar los cálculos en estos casos? La regla es siempre que el orden de los cálculos sea de dentro hacia afuera. A efectos matemáticos es como si estuviéramos utilizando paréntesis, por lo que para hacer el cálculo de un bloque siempre hay que resolver los bloques internos que actúan como operadores.

3. Actividad guiada



Actividad guiada

Operadores

El objetivo de esta actividad guiada es empezar a utilizar bloques de operadores e ir cogiendo soltura en el desarrollo de animaciones.

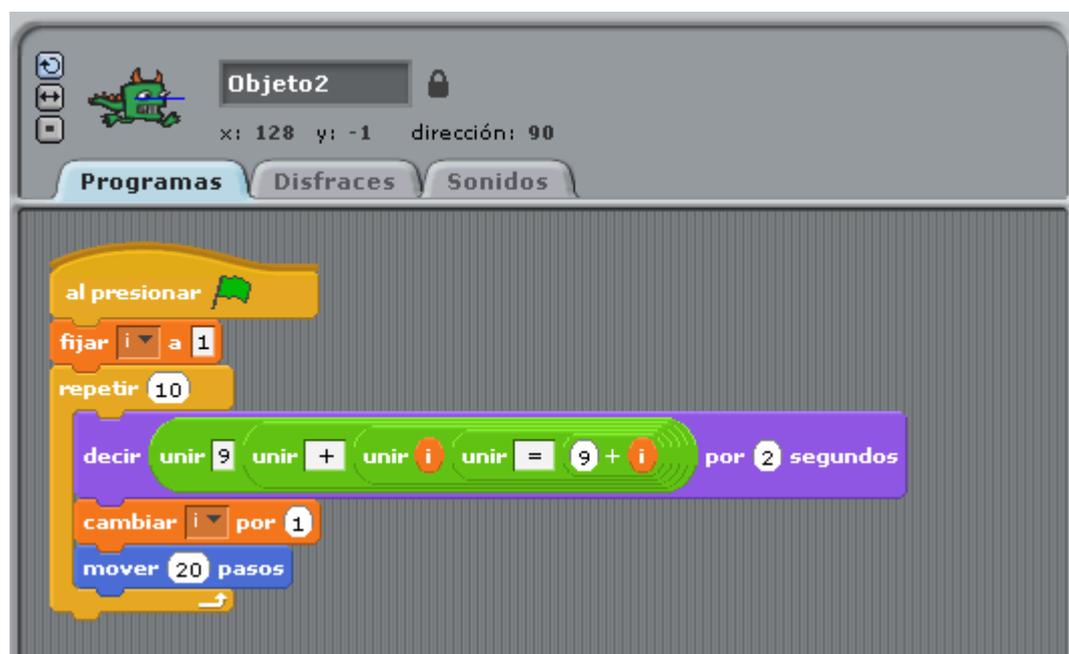
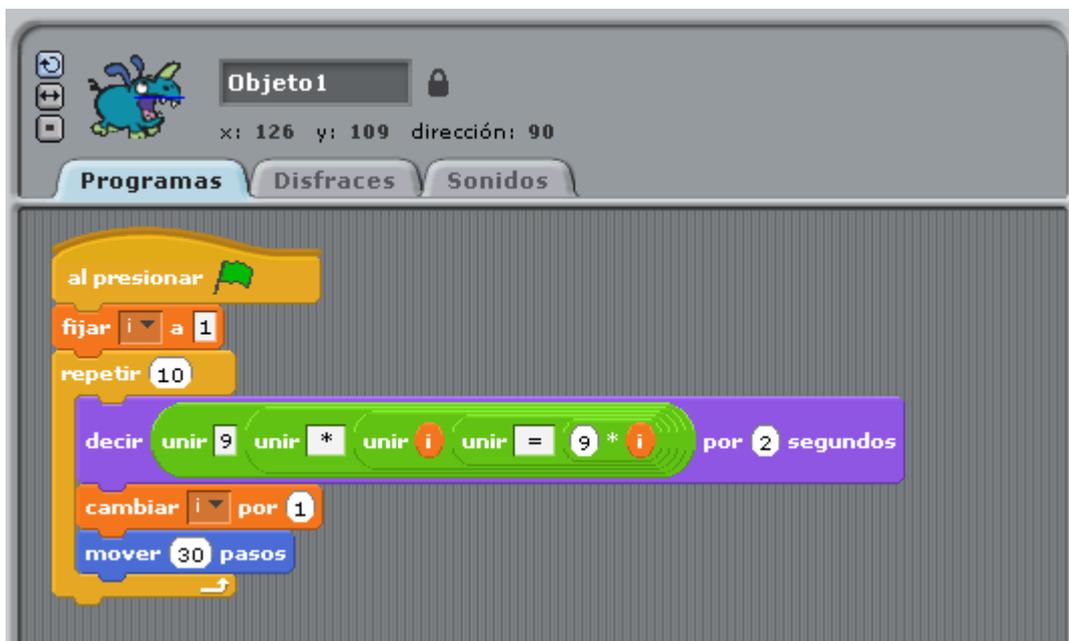
PASO 1 - INICIALIZACIÓN

Creamos un proyecto nuevo en Scratch. Necesitaremos dos objetos para realizar la actividad, utiliza el disfraz que tu quieras para los objetos, los aquí indicados son sólo ejemplos.



PASO 2 - AÑADIENDO CÓDIGO A LOS OBJETOS

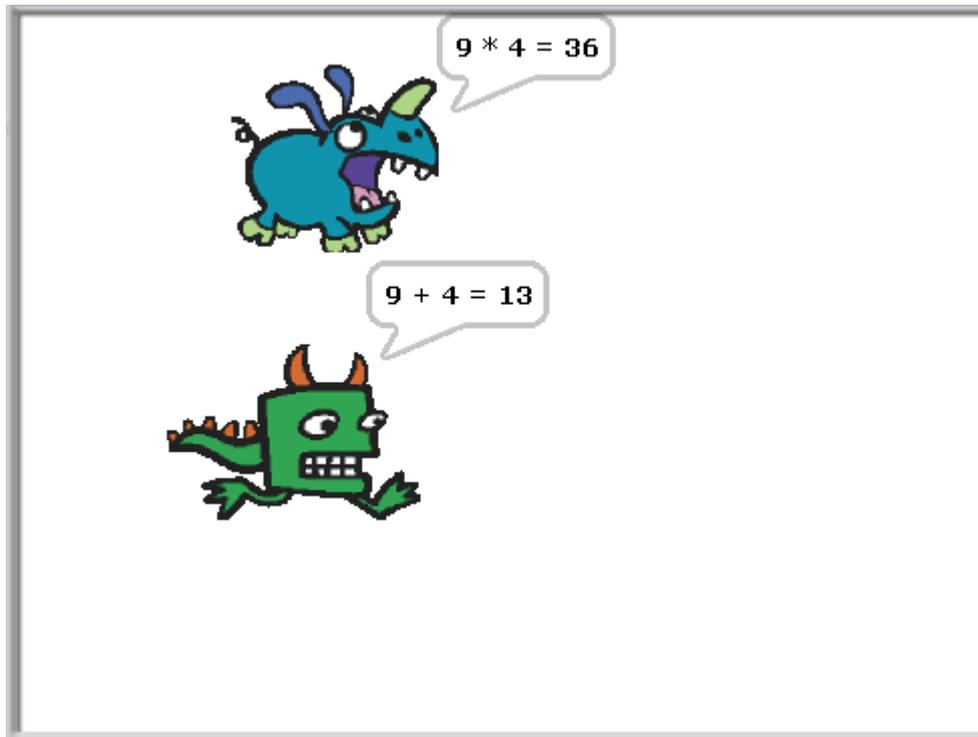
Añade el siguiente código a los objetos del proyecto, **ten en cuenta que la variable i utilizada en ambos casos es local** (lo pregunta en el momento de crear las variables).



Ambos bloques de código son muy parecidos, utilizan una variable i para realizar una pasada de 9 números y realizar cálculos con distintos operadores. En ambos el objeto va moviéndose y va diciendo los cálculos que está realizando.

PASO 3 - COMPROBACIÓN DEL PROGRAMA

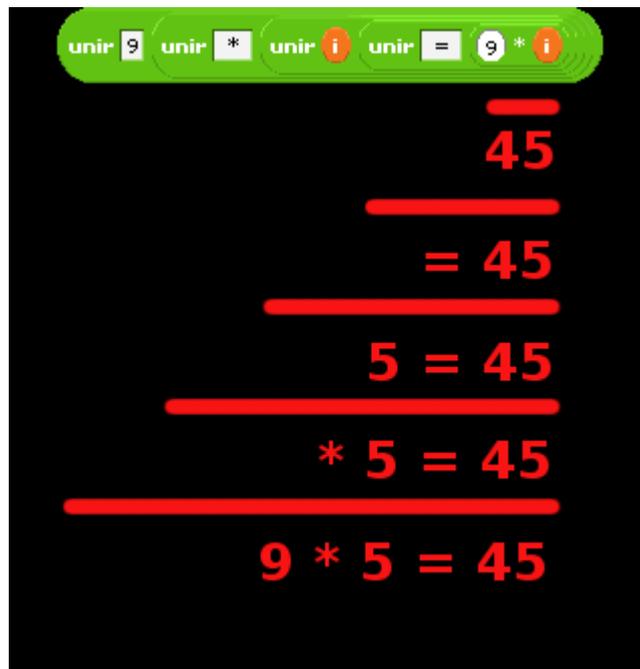
La ejecución del programa debe de mostrar a los dos objetos moviéndose y diciendo las operaciones que van realizando:



Es importante entender como funciona el bloque donde se realizan los cálculos:



Recordad siempre que los bloques anidados se resuelven de dentro hacia afuera. En este bloque tiene un operador aritmético que resuelve la operación y el resto son concateciones de cadenas de caracteres.



Guardad el proyecto como **ActGuiadaOpe** y subela a la plataforma.

4. Fuentes de información



Fuentes de información

- [Página de soporte de la página oficial de Scratch](#)
-