

B
A
S
I
C
O

Scratch: Variables. Operadores aritméticos y lógicos

Duración aproximada: 2 sesiones.



Objetivos

- Conocer el bloque de datos. Aprender a definir y usar las variables.
- Aprender a realizar operaciones matemáticas con variables.
- Conocer los operadores lógicos.
- Utilizar los operadores lógicos y comparaciones.



Fuentes de información

- Scratch. Programación visual. <https://scratch.mit.edu>
- Libro "Programación en LOGO". Autor: Joaquín D'Opazo Álvarez/Grupo Golem. Ed. Anaya Multimedia.
- Aprende Scratch. Sergio Palay. <http://aprendescratch.com>
- © Ángeles Uclés Rodrigo. Autorizado por la autora.



Contenidos

1. Datos. Variables

Los programas manejan datos continuamente y pueden ser de muchos tipos: números, letras, textos, etc. Estos datos se denominan **variables** y su valor cambia a lo largo de la ejecución del programa.

Scratch permite definir variables dentro de la pestaña Programas, bloque Datos:



1.1. Declaración de variables

Para declarar una variable, seguiremos los siguientes pasos:

- Hacer clic en el botón Crear una variable.

Variable nueva

Nombre de la variable:

☒ Para todos los objetos ☐ Sólo para éste objeto

- En el campo, escribir el nombre que queremos para la variable y en los botones de opción marcar si la vamos a usar para todos los objetos (accesible desde todos) o sólo para el objeto actual.
- Por último, hacer clic en el botón OK.

1.2. Asignación de variables

Puede asignarse cualquier valor a una variable mediante los bloques:



- fijar "var_nombre" a "var_valor". Asigna a la variable "var_nombre" el valor "var_valor", donde "var_valor" representa cualquier valor numérico o cadena de texto.
- cambiar "var_nombre" por "var_valor". Suma a la variable "var_nombre" el valor "var_valor", donde "var_valor" representa un valor numérico.

1.3. Mostrar variables

Puede mostrarse u ocultarse el valor de una variable en pantalla mediante los bloques:



2. Operaciones matemáticas

Ahora que ya sabemos cómo declarar y asignar valor a variables, vamos a ver qué operaciones podemos realizar con ellas. Cuando se trata de trabajar con números, Scratch tiene las operaciones habituales en casi todos los lenguajes de programación. Dichos operadores se encuentran dentro del bloque Operadores.

2.1. Operaciones básicas

Corresponden a los símbolos +, -, * y /. Se usan, respectivamente, para la suma, resta, multiplicación y división.



Para la división hay un operador adicional **MOD**, que devuelve el resto del resultado de la división. Por ejemplo: $9 \text{ MOD } 4 = 1$

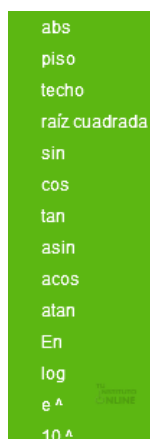


El bloque redondear, redondea un número al entero más próximo.



2.2. Operaciones adicionales

Aparte de estos operadores, existen funciones matemáticas adicionales para realizar calculos más complejos. En el bloque correspondiente, desplegamos la lista de funciones adicionales (valor absoluto, funciones trigonométricas, logaritmos, potencias, raíces cuadradas, etc.):



También tenemos un bloque para **generar un número entero al azar** entre 2 valores:

número al azar entre 1 y 10

3. Ejercicio práctico

A continuación vamos a realizar un ejercicio práctico de repaso de todos los contenidos vistos anteriormente.



Ejercicio

3.1. Sumar 2 números

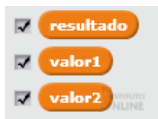
Vamos a crear un programa que sume dos números. Aparecerá un personaje que te pedirá dos números y luego te dará el resultado.

Crear y guardar nuevo proyecto

- Crea un nuevo programa.
- Ve al menú Archivo → Guardar como (local). Pon como nombre "pryopsbasicas" y guárdalo en tu carpeta de "SCRATCH".

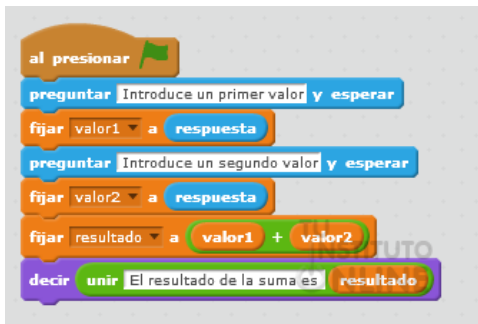
Crear variables

- Primero, crea tres variables:



Sumar 2 números

- **Ayuda.** El código del personaje será:



Multiplicar 2 números

- Modifica el programa para que después de sumar los 2 números, los multiplique y devuelva el resultado por pantalla.

Subir el proyecto

- **Sube el fichero "pryopsbasicas.sb2" a la plataforma.**



Contenidos

4. Comparaciones

Además, también disponemos de los operadores $<$, $>$, $=$ para efectuar comparaciones lógicas.



Operador	Significado
<	Compara si un valor es menor que otro.
>	Compara si un valor es mayor que otro.
=	Compara si un valor es igual que otro.

5. Operadores lógicos

La palabra “booleano” viene del álgebra desarrollada por Boole. Los datos de tipo booleano sólo pueden tomar 2 valores: VERDADERO o FALSO. También se denomina a estos datos como tipo lógico. Las operaciones de valores booleanos con operadores de tipo booleano forman las expresiones booleanas y se usan para tomar decisiones lógicas.

Para realizar operaciones entre expresiones cuyo resultado sea Boolean, existen algunas instrucciones similares a las que podemos ver en casi todos los lenguajes de programación. Se trata de Y, O y NO.

5.1. Operador Y

Corresponde al operador Y lógico. Su tabla de verdad que lo define es:

A	B	A y B	Ejemplo
F	F	F	Si A=5 y B>7 entonces...
F	T	F	
T	F	F	
T	T	T	
T	T	T	

5.2. Operador O

Corresponde al operador O lógico. Su tabla de verdad que lo define es:

A	B	A o B	Ejemplo
F	F	F	Si A = 5 o B < 8 entonces...
F	T	T	
T	F	T	
T	T	T	
T	T	T	

5.3. Operador NO

Corresponde al operador NO lógico. Su tabla de verdad que lo define es:

A	NO A	Ejemplo
F	T	Si $\text{NO}(A=7)$ entonces...
F	T	
T	F	
T	F	

6. Ejercicios prácticos

A continuación vamos a realizar dos ejercicios prácticos de repaso de todos los contenidos vistos anteriormente.



Ejercicio

6.1. Adivinar un número

Una de las primeras posibilidades de las funciones de números aleatorios es implementar el programa de adivinar un número. Vamos a empezar con la versión sencilla del juego: el ordenador, al azar, piensa un número entero entre 1 y 100. El usuario teclea un número y el programa nos dice si hemos acertado o no.

Crear y guardar proyecto

- Crea un nuevo programa.
- Ve al menú Archivo → Guardar como (local). Pon como nombre "pryadivinanum" y guárdalo en tu carpeta de "SCRATCH".

Variables

- Define una variable (por ejemplo NUM_AZAR) para almacenar el número generado al azar entre 0 y 100. Muestra la variable.

Programa. Instrucciones

- Genera un número al azar entre 1 y 100. Asigna el número generado a la variable NUM_AZAR.
- Escribe un texto pidiendo un número al jugador.
- Lee un número por teclado.
- Ahora comparamos nuestro número con el del ordenador.
 - Si coincide con el número generado al azar, hemos acertado. Se muestra un mensaje de enhorabuena y toca un sonido de "clapping" o "cheer". Espera 2 segundos.
 - En caso contrario, muestra un mensaje del tipo "has perdido" y toca un sonido "goose". Espera 2 segundos.
- Para finalizar, muestra un mensaje con el número generado al azar por el ordenador.
- **Ayuda:**



- Ejecuta el programa y juega a adivinar el número.

6.1.1. Ampliar número de intentos

El programa, tal y como lo hemos implementado, no tiene mucho sentido, ya que es muy difícil adivinar un número al azar entre 1 y 100 con una sola oportunidad. Teniendo en cuenta la probabilidad matemática, para este caso necesitamos disponer de al menos 6 intentos para adivinarlo.

6.1.2. Indicar si el número es mayor o menor

Aunque hemos añadido 6 oportunidades, adivinar el número sigue siendo como buscar una aguja en un pajar. Vamos a hacer que en cada intento, el ordenador responda si es menor, mayor o hemos acertado.

Programa. Instrucciones

Añade un bucle con 6 repeticiones para que podamos jugar las 6 oportunidades. Vamos a eliminar la instrucción condicional SI...NO y la sustituiremos por 3 condicionales simples.

- Repetir 6 veces:
 - Escribe un texto pidiendo un número al jugador.
 - Lee un número por teclado.
 - Ahora comparamos nuestro número con el del ordenador.
 - Si es mayor, muestra un texto indicándolo y espera 2 segundos. Continuamos con el siguiente intento.
 - Si es menor, muestra un texto indicándolo y espera 2 segundos. Continuamos con el siguiente intento.
 - Si coincide con el número generado al azar, hemos acertado. Se muestra un mensaje de enhorabuena y toca un sonido de "clapping" o "cheer". Espera 2 segundos.
- Después del bucle repetir:
 - Para finalizar, muestra un mensaje con el número generado al azar por el ordenador.
- Ayuda:



- Ejecuta el programa y juega a adivinar el número.
- ¿Qué ocurre si acertamos el número? El programa no se detiene hasta que se completan los 6 intentos. Este funcionamiento no es correcto, puesto que si hemos acertado, el programa debe pararse.

6.1.3. Corrección de errores

Programa. Instrucciones

- Añade el bloque **detener todos** cuando se acierte el número.
- Si agotamos el máximo número de intentos (6), finalizaremos el bucle y ejecutaremos las instrucciones que están justo después. Por ello, además de decir el número que había pensado el ordenador, debes añadir: muestra un mensaje del tipo "has perdido" y toca un sonido "goose". Espera 2 segundos.
- Oculta la variable NUM_AZAR para que no se vea en pantalla el número generado al azar por el programa.
- Juega a adivinar el número y comprueba que ahora funciona correctamente.

Subir el proyecto

- **Sube el fichero "pryadivinanum.sb2" a la plataforma.**



Ejercicio

6.2. Lanzar un dado

Vamos a implementar un programa que lance un dado y nos diga el resultado.

Crear y guardar proyecto

- Crea un nuevo programa.
- Ve al menú Archivo → Guardar como (local). Pon como nombre "prydado" y guárdalo en tu carpeta de "SCRATCH".

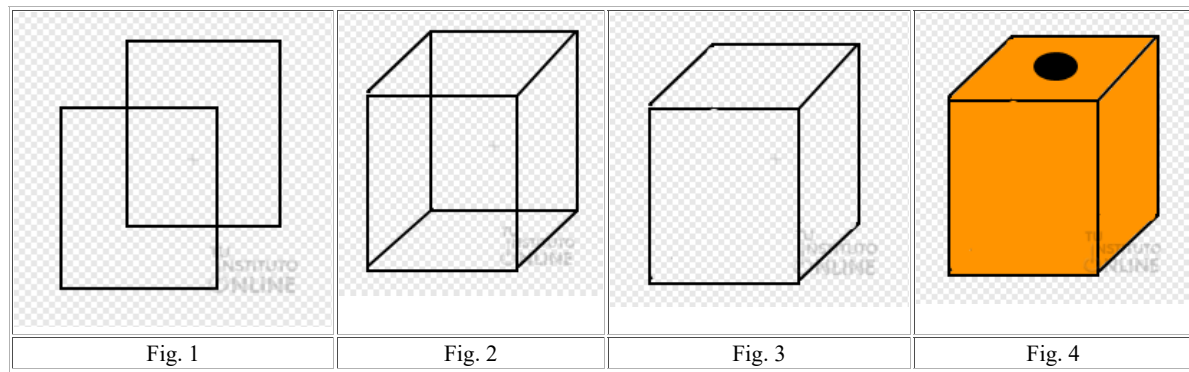
Crear objetos

En primer lugar, vamos a dibujar el dado.

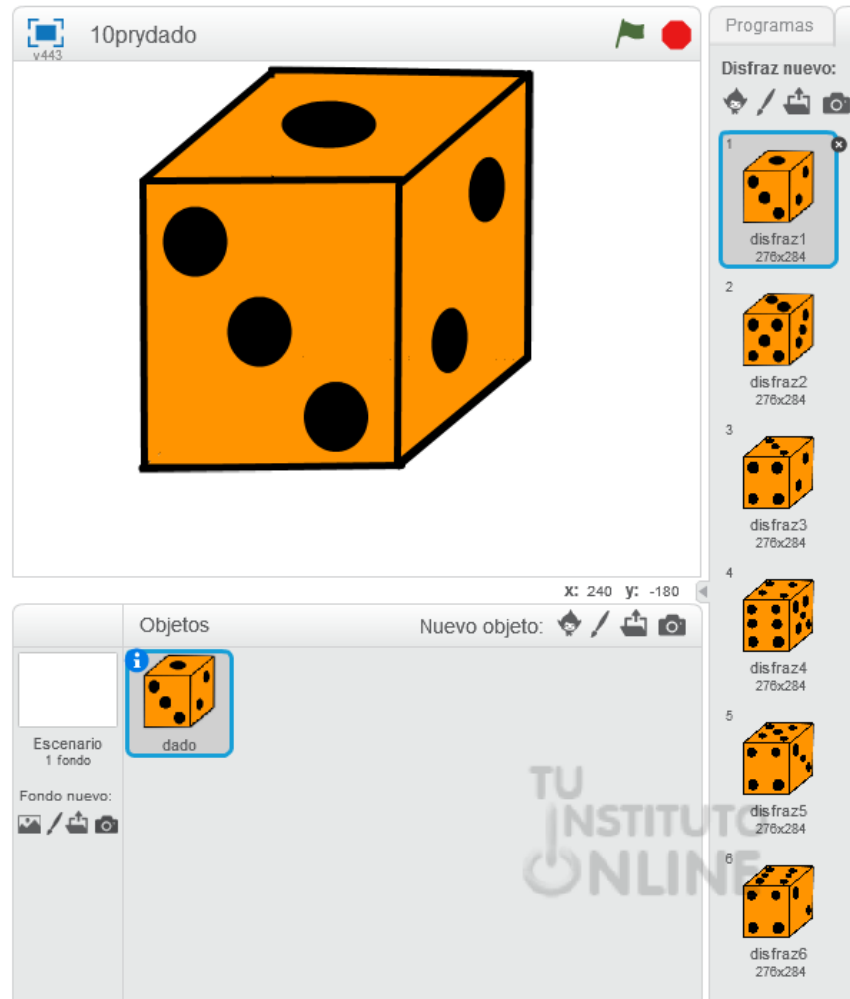
- Crea un nuevo objeto. Renombra el objeto como "dado".
- Crea un nuevo disfraz. Renombra el disfraz como "disfraz1". Ve a la pestaña Disfraces, icono Pintar nuevo disfraz.



- Dibuja dos cuadrados (fig.1). Une los vértices (fig.2) y, por último, borra las aristas que no se necesiten (fig.3). Por último, pinta en la cara superior un punto y completa las otras caras (fig.4).

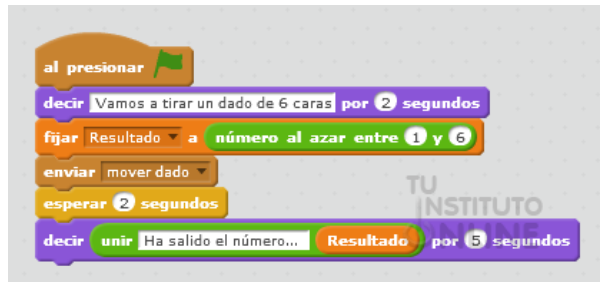


- Después, duplica el primer disfraz seis veces, de modo que el disfraz 2 tenga dos puntos en la cara superior, el disfraz 3, tres puntos y así sucesivamente.



Programa. Instrucciones

- Presenta el evento con una frase.
- Crea una variable llamada "resultado" para almacenar el resultado de lanzar el dado. El valor va a ser un número entre 1 y 6.
- Después, manda un mensaje al dado para que se mueva.
- Esperamos un tiempo determinado y, al final, decimos el resultado que ha salido.



Programación del objeto dado

En el caso del dado, vamos a hacer que se mueva cuando recibe el mensaje de mover dado.

- Establece una posición por defecto en la pantalla.
- Coloca el dado en posición horizontal.
- Por defecto, comienza con el disfraz 1.
- Luego, vamos a repetir una cantidad determinada de veces (el 11 sale para que el dado dé dos vueltas enteras antes de mostrar el resultado, si quisiéramos una vuelta sola sería 5, y si no, sólo resultado). Los bloques de mover y girar le dan un efecto de movimiento, pudiendo ponerse los valores que se deseen. El cambio de disfraz simula que el dado va dando vueltas, ya que cambian las caras.
- Por último, nos aseguramos de que el dado siempre termine en posición horizontal.
- **Ayuda:**



- Ejecuta el programa y comprueba que funciona correctamente.
- Guarda los cambios.

Subir el proyecto

- **Sube el fichero "prydado.sb2" a la plataforma.**