INSTITUCIÓN EDUCATIVA JORGE CLEMENTE PALACIOS.
TIBASOSA –BOYACÁ

ÁREA: TECNOLOGÍA E INFORMÁTICA

GUIA DE APRENDIZAJE NIVELACION 3. 2025
DOCENTE: Fredy Alexander Bello Caicedo
ASIGNATURA: INFORMATICA
GRADOS: 10
HORAS DE CLASES:
FECHA:
TEMA: Programación por bloques,
LOGRO: Reconocer la importancia de los entornos de programación, como medio de expresión y desarrollo de un pensamiento algorítmico.

Guía N.4

RECORDEMOS: DESCRIPCION DE BLOQUES

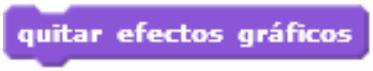
los bloques de Scratch están organizados dentro de 8 categorías de códigos de color:

1. **Movimiento.**
2. Apariencia.
3. Sonido
4. **Lápiz**
5. Control
6. Sensores
7. Operadores
8. variables

LA APARIENCIA DE LOS OBJETOS

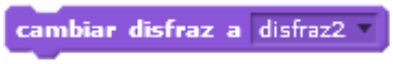
Los objetos pueden cambiar su apariencia, lo que incluye mostrarse, esconderse, cambiar de tamaño, de color, brillo, etc.

Los objetos también, pueden cambiar de disfraz. Los disfraces de un objeto son las diferentes imágenes con las que puede mostrarse, pero la

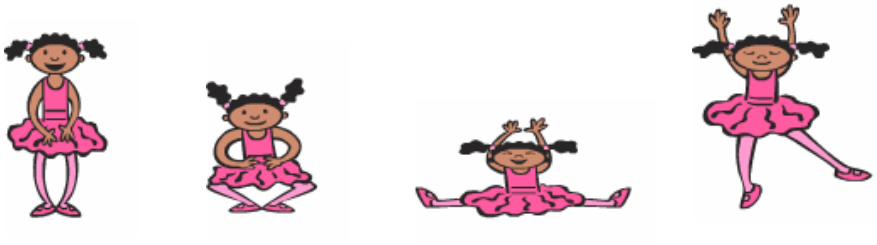
	Hace aparecer el objeto en el escenario		Hace desaparecer el objeto del escenario. Un objeto escondido no puede ser detectado por otros objetos con el bloque 
		Establece el efecto gráfico seleccionado (color, brillantez, desvanecer, etc.) en el valor indicado.	
		Cambia (aumenta si valor positivo o disminuye si valor negativo) el efecto seleccionado en el valor indicado.	
		Quita los efectos gráficos anteriores que tenga aplicados (afecta al color, la brillantez, etc., pero no al tamaño	
		Establece el tamaño del objeto al valor indicado.	
		Aumenta (valores positivos) o disminuye (valores negativos) el tamaño del objeto en % el valor indicado.	

diferencia entre disfraces no tiene que limitarse a cambios de tamaño o color, sino que puede tratarse de diferentes posiciones de un mismo personaje, para dar la sensación de movimiento al ir cambiando con rapidez una por otra (como los GIF animados) o incluso tratarse de imágenes diferentes.

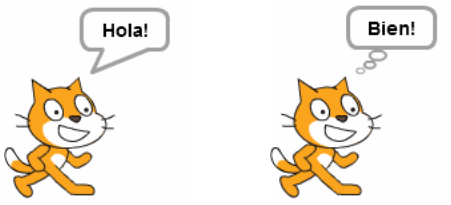
Por ejemplo, el gatito tiene dos disfraces. Al cambiar rápidamente uno por otro parece que camina. La bailarina tiene cuatro y al ir cambiando uno por otro rápidamente parece que baila.

	Cambia el disfraz actual por el seleccionado.
	Cambia el disfraz actual por el siguiente de la lista de disfraces. Cuando llega al último vuelve al primero.

Ejemplo: Haz que la bailarina baile:

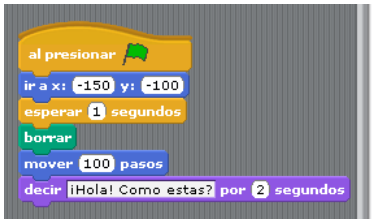


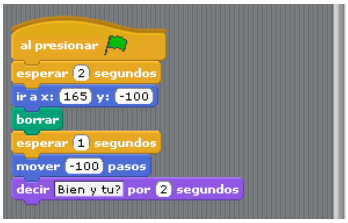
Los objetos también pueden emitir mensajes en forma de burbujas de diálogo o de pensamiento.



	Despliega una nube de diálogo o de pensamiento del objeto de forma permanente. Esta nube se elimina si se ejecuta el bloque sin texto.
 	Despliega la nube de diálogo o de pensamiento por el tiempo indicado.

Ejemplo: Encuentro de Scratch y cangrejo.

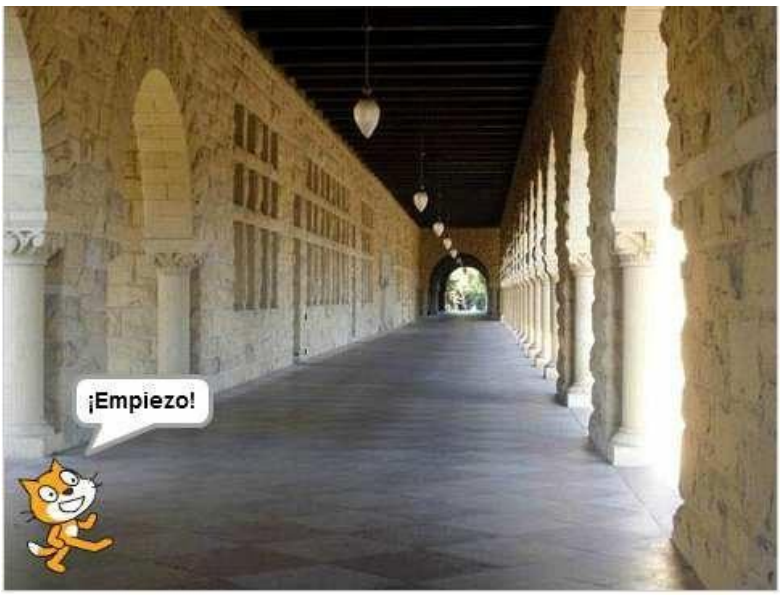


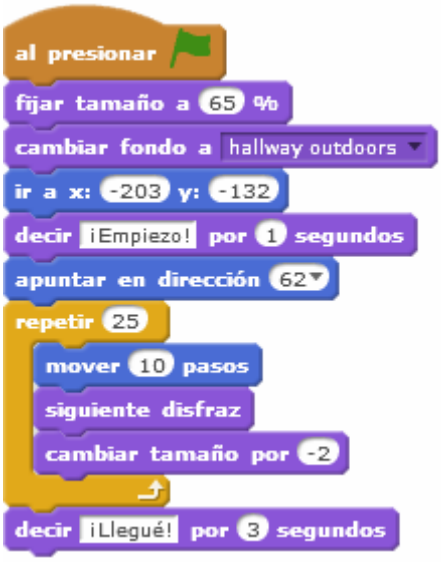


El **escenario** es, prácticamente, como un objeto más: puede tener programas asociados; puede tener varios disfraces, que en este caso se denominan “fondos”; se le pueden aplicar efectos gráficos como efectos de color, de brillo, etc. Para añadir nuevos fondos al escenario podemos usar varias herramientas, desde cargarlos desde la biblioteca, a cargarlos desde un archivo de imagen o pintarlos nosotros mismos.

Ejemplo :

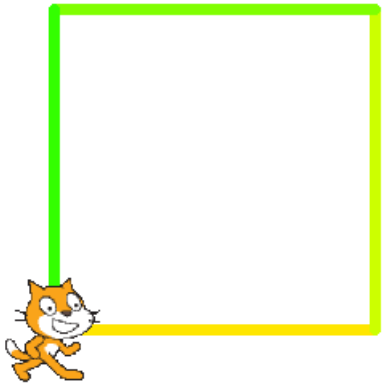
Carga desde la biblioteca el fondo llamado “hallway outdoors” y, a continuación, construye y ejecuta para el gatito el programa de la figura. Observarás que el gato empieza al 65% de su tamaño en la esquina inferior izquierda del escenario, dice el mensaje “¡Empiezo!” y empieza a moverse cambiando de disfraz en dirección 62º al mismo tiempo que se va reduciendo su tamaño para parecer que se está alejando. Al final del trayecto emite el mensaje “¡Llegué!”. En cuanto al escenario, aunque inicialmente estemos con el fondo blanco por defecto, el propio programa del gato hace que cambie el fondo al que le indicamos.



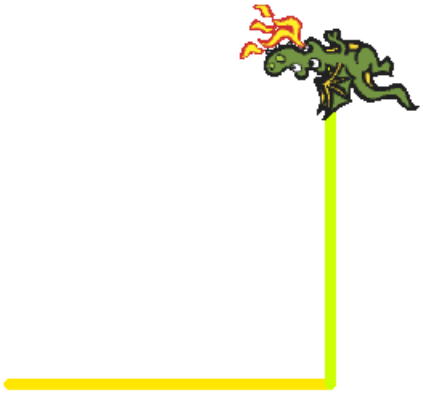


Actividad de Nivelación 1:

1. Diseñe un programa para que el gatico, describa una trayectoria cuadrada.



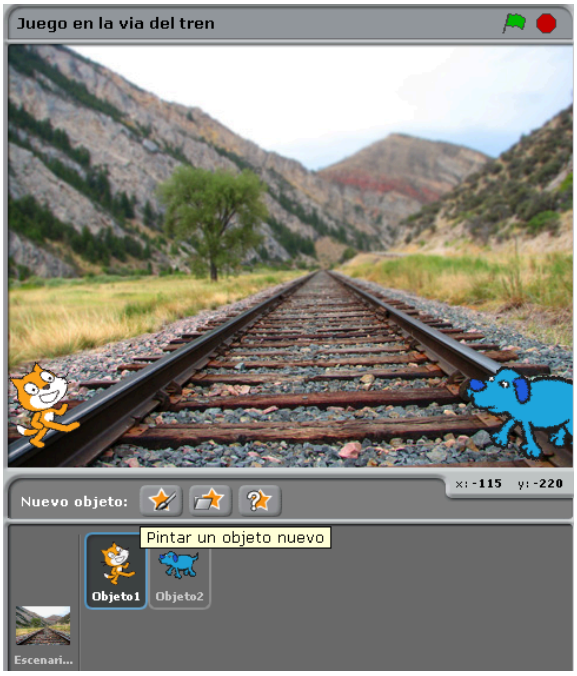
2. Que los lados del cuadrado cambien de color.
3. Que al llegar a cada vértice cambie de disfraz.



4. Que cuando cambie de disfraz emita un sonido.

Actividad de Nivelación 2:

1. Scratch y su amigo están jugando un juego muy peligroso en la vía del tren, Van y vuelven corriendo haber quien es el más rápido.
2. Diseñe un programa para que el gatico y su amigo, reduzcan su tamaño a medida que se alejan y aumente su tamaño a medida que se acercan.
3. Algunas veces gana Scratch y algunas veces su amigo.
4. Inserten el fondo llamado **train –tracks 1**



Guía N.5

RECORDEMOS: DESCRIPCION DE BLOQUES

los bloques de Scratch están organizados dentro de 8 categorías de códigos de color:

1. Movimiento.
2. Apariencia.
3. Sonido
4. Lápiz
5. Control
6. Sensores
7. Operadores
8. variables

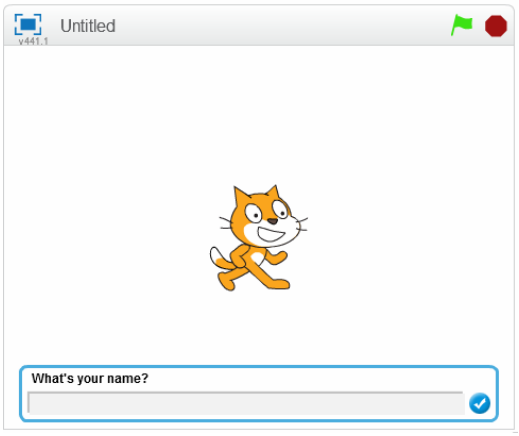
LA ENTRADA DE DATOS DESDE EL TECLADO

En muchas ocasiones necesitamos suministrarle datos a los programas. Por ejemplo, podemos pensar en un programa que nos dibuja polígonos regulares, y nos tiene que preguntar el número de lados y la longitud del lado del polígono.

Los objetos (incluido el escenario) pueden hacer preguntas al usuario para que éste las introduzca a través del teclado.

preguntar What's your name? y esperar	Formula una pregunta en la pantalla y guarda lo que introduzcamos en respuesta. Hace que el programa espere hasta presionar "Enter" o hacer clic en la casilla de verificación.
respuesta	Almacena la entrada por teclado más reciente tras la ejecución del bloque anterior por parte de cualquier objeto o del escenario. Es decir, es compartido por todos los objetos.

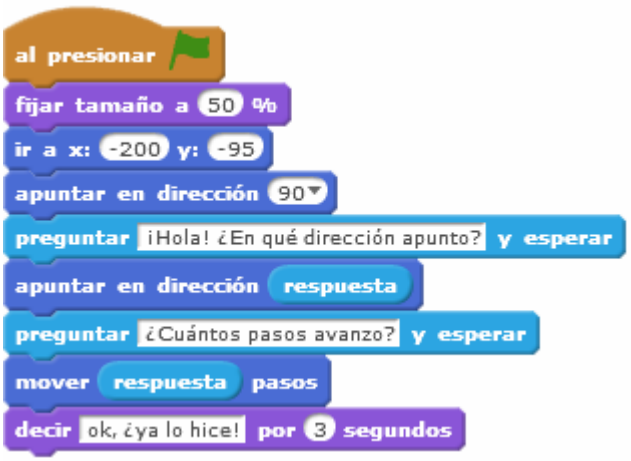
Cuando es un objeto el que formula la pregunta ésta aparece en una nube de diálogo, mientras que aparece en la parte inferior de la pantalla cuando pregunta el escenario.



Ejemplo 1:

Construye y ejecuta para el gatito el programa de la figura. Observa como la "respuesta" a las preguntas se puede introducir en los huecos destinados a los datos que hay en los bloques.

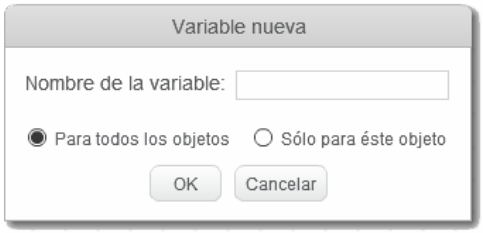
Observa también como tras la primera pregunta, el dato que hay en el bloque "respuesta" es el valor del ángulo de dirección que hayamos introducido, pero después de la segunda pregunta lo que hay en "respuesta" es ya el número de pasos que debe moverse. O sea, solo se guarda el valor más reciente.



EL USO DE LAS VARIABLES

Una variable podemos decir que **es un espacio en la memoria con un nombre**. En dicho espacio podemos guardar un dato, que puede ser un número o una cadena de caracteres (palabras o frases).

Para crear **variables** abrimos el grupo de bloques Datos y hacemos clic sobre Crear una **variable**. En el cuadro de diálogo que se abre indicamos el nombre de la variable y seleccionamos si queremos que sea accesible por todos los objetos o solo para el objeto activo en el momento actual.

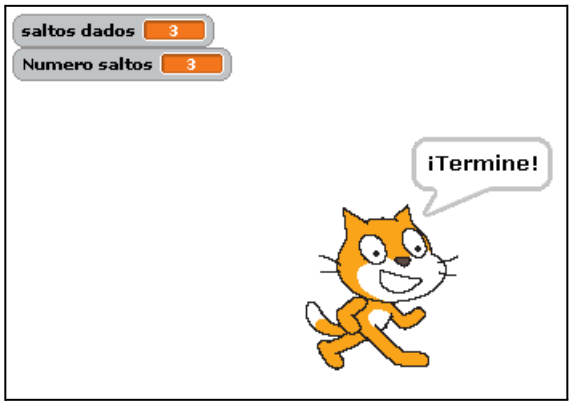


Dato	Contiene el dato guardado en la variable y podemos usar este bloque dentro de los huecos redondeados o rectangulares para ingreso de información de otros bloques.
Dato	Clicando sobre la casilla de verificación podemos monitorizar o no en pantalla el valor actual de la variable. Puede activarse o desactivarse durante la ejecución del programa.
fijar Dato a 0	Establece el valor de la variable seleccionada en el menú desplegable al valor indicado.

	Incrementa (si el valor es positivo) o disminuye (si el valor es negativo) el valor de la variable seleccionada en el número indicado.
	Muestra u oculta el valor de la variable en el escenario desde el programa en ejecución.

El valor de la variable se puede monitorizar en pantalla con diferentes formatos. Se cambia de uno a otro haciendo doble clic o haciendo clic con el botón derecho del ratón y seleccionado el formato.

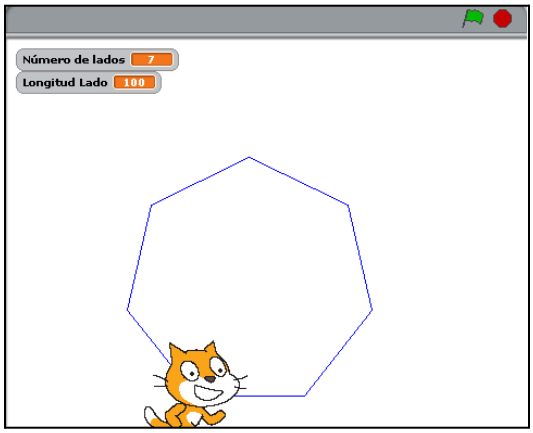
Ejemplo 2: El gato pregunta cuántos saltitos debe dar y guarda el dato en la variable "Número saltos". A continuación repite tantas veces como indique el número introducido el conjunto de bloques que producen el efecto del salto. Cada vez que da un salto dice el número del salto dado, el cual se guarda en la variable "Saltos dados", la cual se incrementa en 1 con cada salto. Al finalizar dice ¡Terminé!



Ejemplo 3: Programa que dibuja polígonos regulares. Para ello, nos pide el número de lados y la longitud del lado del polígono. Ten en cuenta que si la longitud del lado que introducimos es demasiado grande el polígono puede no entrar en la pantalla y se realiza mal.

Hemos creado dos variables: "Número lados" y "Longitud lado".

Observa que hemos utilizado un **operador matemático**, el de la división, que lo veremos en breve.



Actividad de Nivelación 3.

1. Diseñe un programa que pregunte cuantas canastas quiere que anote el gato, que lance el balón y anote las canastas.

Guía N.6

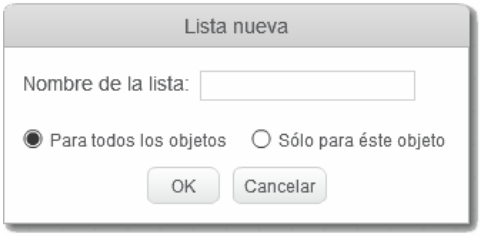
RECORDEMOS: DESCRIPCION DE BLOQUES

los bloques de Scratch están organizados dentro de 8 categorías de códigos de color:

1. Movimiento.
2. Apariencia.
3. Sonido
4. Lápiz
5. Control
6. Sensores
7. Operadores
8. variables

LAS LISTAS

Una lista es una variable múltiple (contiene varias variables simples). Es como si fuera un cajón con varios compartimentos. Cada compartimento es una variable simple y en él se puede guardar un dato (un número o una cadena de caracteres). Todas las variables simples tienen el mismo nombre (el que le demos a la lista) y se diferencian por un número que indica el lugar que ocupan en la lista (a este número se le llama índice).



Para crear una lista se procede igual que con las variables, abrimos el grupo de bloques Variables y hacemos clic sobre Crear una lista, y damos su nombre y para quién es accesible. Tener cuidado a elegir los nombres de las listas pues, a diferencia de las variables simples, posteriormente no podremos cambiarlos.

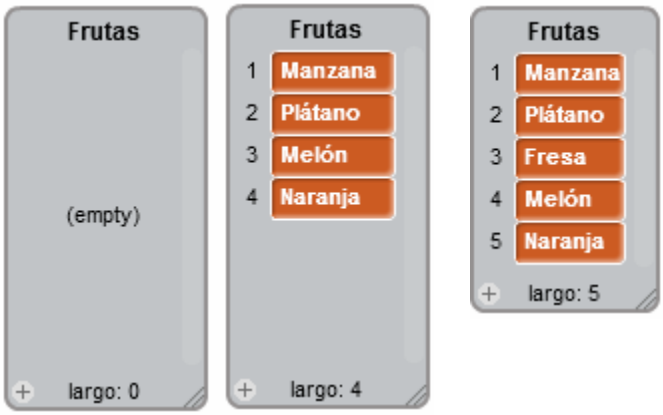
Datos	Contiene todos los datos guardados en la lista y podemos usar este bloque dentro de los huecos redondeados o rectangulares para ingreso de información de otros bloques.
elemento 1 de Datos	Contiene el elemento indicado en el hueco de la lista seleccionada. Puede incluirse en los huecos de otros bloques.
Datos	Clicando sobre la casilla de verificación podemos monitorizar o no en pantalla el contenido actual de la lista. Puede activarse o desactivarse durante la ejecución del programa.
añade thing a Datos	Añade lo que incluyamos en el hueco como un elemento adicional que se coloca en el último lugar de la lista seleccionada.
insertar thing en 1 de Datos	Inserta lo incluido en el hueco como el elemento indicado de la lista. Los elementos con este índice o superior se desplazan hacia delante.
borrar 1 de Datos	Borra el elemento cuyo índice es el número indicado de la lista seleccionada. Los elementos de índice superior al indicado se desplazan hacia atrás en la lista
reemplazar elemento 1 de Datos con thing	Reemplaza el elemento indicado de la lista seleccionada por lo que incluyamos en el hueco.
mostrar lista Datos / esconder lista Datos	Muestra u oculta el contenido de la lista en el escenario desde el programa en ejecución.
longitud de Datos	Contiene el número total de elementos que tiene actualmente la lista seleccionada.
Datos contiene thing ?	Este bloque proporciona un valor lógico “verdadero” si la lista contiene algún elemento que coincida exactamente con lo indicado en el hueco.

Los bloques anteriores permiten modificar (añadir, borrar, insertar, reemplazar), consultar y mostrar o esconder las listas desde los programas. Por ejemplo, programas que van construyendo listas a partir de unos datos que va suministrando el usuario. No obstante, si nuestro programa tiene que partir de unas listas ya

construidas desde el momento inicial, podemos hacerlo desde el monitor de lista que podemos mostrar en el escenario.

En el momento de crear la lista, este monitor aparecerá vacío y con longitud 0. Podemos hacer:

- Si le damos al icono + situado en la esquina inferior izquierda podemos añadir nuevos elementos.
- Haciendo clic sobre un elemento y clic en la X lo eliminamos.
- Si seleccionamos un elemento haciendo clic sobre él y a continuación en + insertamos un nuevo elemento tras el que hemos seleccionado.
- Para cambiar el valor de un elemento solo tenemos que seleccionarlo haciendo clic y cambiar el contenido.
- Podemos redimensionar el tamaño del monitor de lista desde la esquina inferior derecha.



Ejemplo 1: Vamos a partir **de una lista ya hecha** con las provincias de Andalucía, pero con varios errores (provincias que no son, otras que faltan, alguna repetida). Haremos un programa que corrija los errores y deje bien la lista y en orden alfabético, practicando algunos de los bloques que hemos visto. Como mostraremos la lista en el escenario podemos ir viendo los cambios conforme se van haciendo, por lo que pondremos un tiempo de espera entre un cambio y otro para que nos dé tiempo de visualizarlos. Debemos tener cuidado pues cada vez que hacemos operaciones de borrado o de inserción, se producen cambios en los índices de los elementos.

Observamos que los errores son:

sobran Cuenca y Segovia,

faltan Cádiz, Huelva y Sevilla,

Granada está repetida.

Las operaciones que haremos serán:

- Insertar Cádiz entre Almería y Córdoba (esto hace que los índices de Córdoba y las que están debajo aumenten en 1).
- Borrar Cuenca (esto hace que los índices de las que están debajo disminuyan en 1).
- Reemplazar Segovia por Huelva.
- Borrar Sevilla de donde está
- Borrar la segunda aparición de Granada.
- Añadir Sevilla al final.

Actividad de Nivelación 4

1. A partir de la siguiente lista ya hecha con algunas veredas del municipio de Tibasosa, y con varios errores (elementos que no son, otros que faltan y algunos repetidos). Realice un programa que corrija los errores y deje bien la lista y deje en orden las primeras 8 veredas. Practicando algunos de los bloques que hemos visto.

Guía N.7

RECORDEMOS: DESCRIPCION DE BLOQUES

los bloques de Scratch están organizados dentro de 8 categorías de códigos de color:

1. Movimiento.
2. Apariencia.
3. Sonido
4. Lápiz
5. Control
6. Sensores
7. Operadores
8. variables

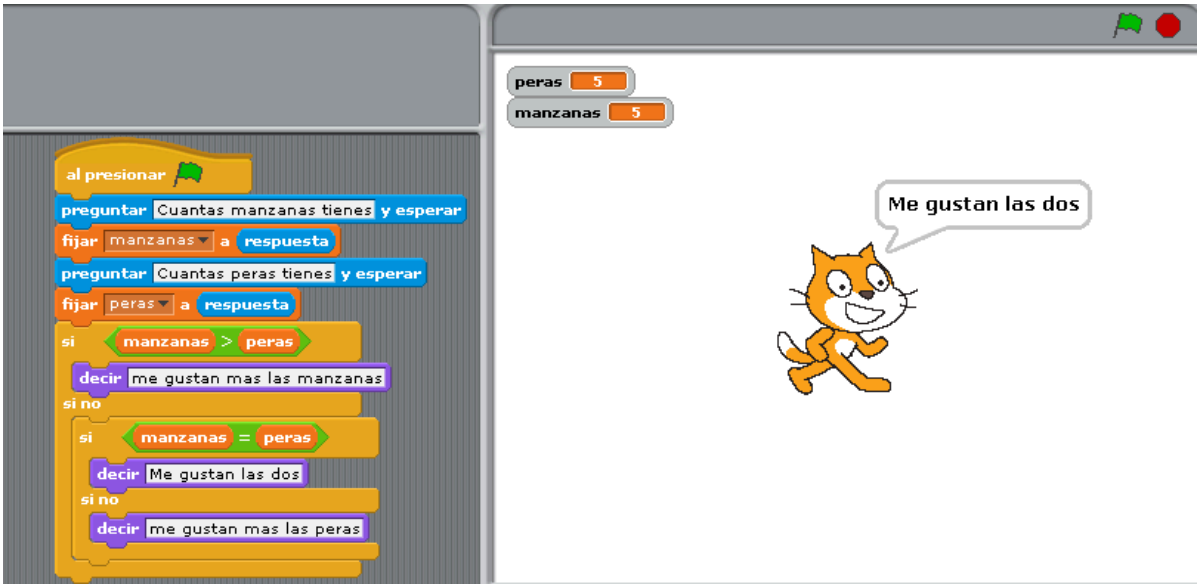
LA TOMA DE DECISIONES

Llegamos a una de las partes más importantes de la programación: la toma de decisiones. Existen unas instrucciones, llamadas "*sentencias condicionales*", que, tras evaluar si una *condición* determinada se cumple o no, deciden ejecutar un bloque de código o no ejecutarlo, o bien deciden ejecutar un bloque de código o ejecutar otro.

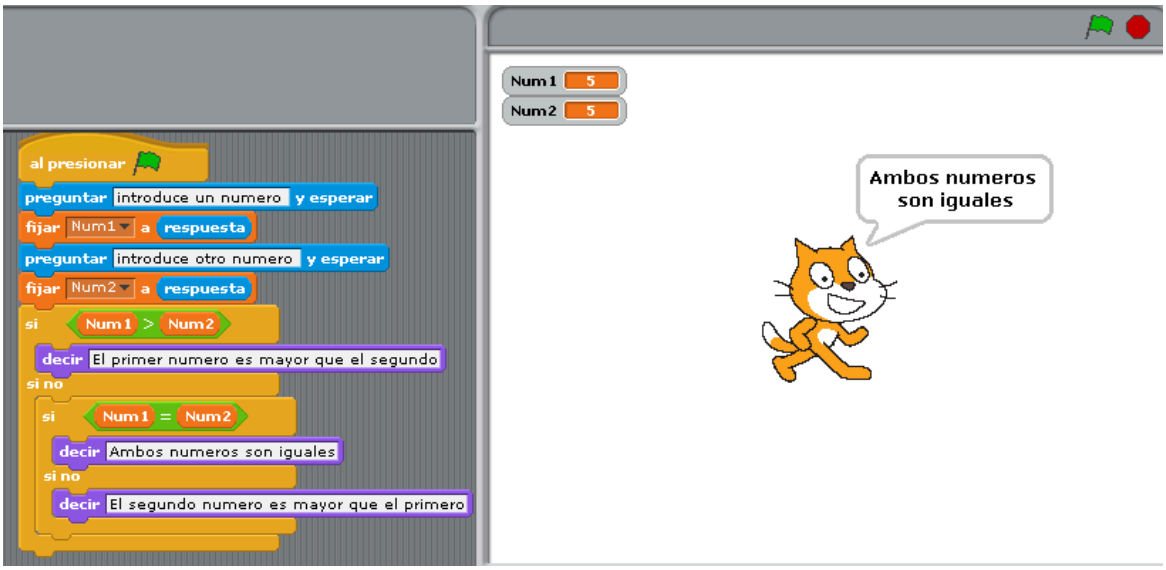
	Sentencia condicional **SI**. Ejecuta el conjunto de bloques de instrucciones que coloquemos en su interior únicamente si la condición que introduzcamos en el hueco es verdadera. La condición puede ser una operación relacional de comparación, o si se presiona una tecla o el ratón, o si un objeto toca a otro, etc.
	Sentencia condicional **SI ….. SI NO**. Si la condición que introduzcamos en el hueco es verdadera ejecuta el conjunto de bloques de instrucciones situado dentro de la abertura de SI, y si la condición es falsa, se ejecuta el conjunto de bloques situado en la abertura de SI NO.

Ejemplo 1. Escribe el programa para poder escribir de qué fruta tienes más, si peras o manzanas.

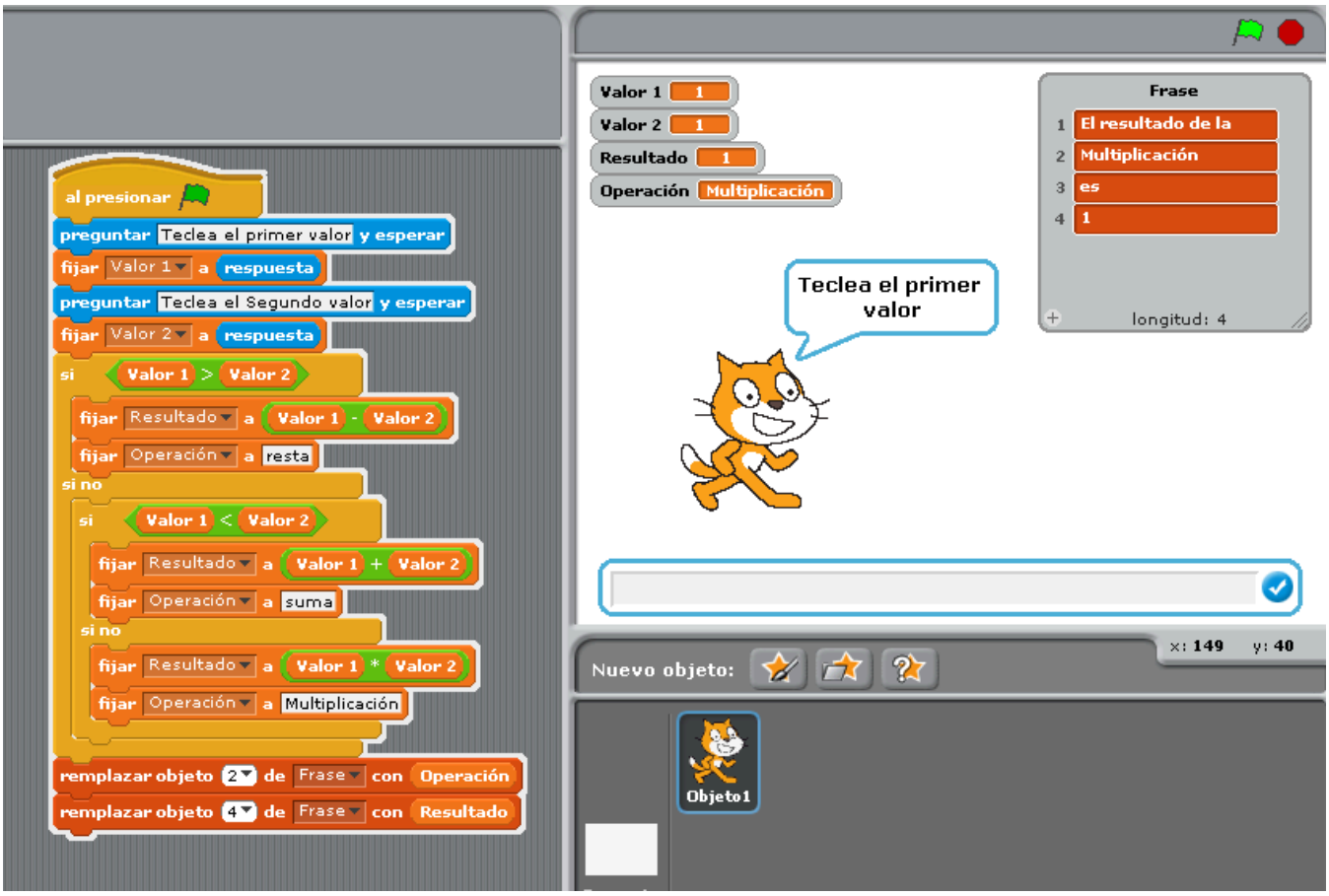
Primero crearemos dos variables: peras y manzanas. Preguntamos cuantas manzanas tienes y Fijamos el valor a Respuesta. Igualmente lo hacemos para peras. Desde la pestaña Control seleccionamos nuestras cajas para trabajar con condicionales. El resultado será:



Ejemplo 2. Decir cuál es el mayor de dos números.



Ejemplo 3: Vamos a diseñar un programa en el que el gatito nos pida dos valores numéricos. En caso de que el primero sea mayor que el segundo, **realizará la resta**. Si el primero es menor que el segundo, realizará la suma y si son iguales los multiplicará. Al final, el gatito dirá la operación realizada y su resultado.
Lo haremos creando una lista, llamada Frase, donde dejaremos dos Huecos para rellenarlos con la operación y con el resultado.



Actividad de Nivelación 5:

1. lea cuidadosamente la guía.
2. Copie la guía en su cuaderno.
3. Practique los ejemplos.
4. Practique cada una de las herramientas que tiene esta categoría.
5. Realice el sig **Ejercicio:** Tengo la nota de mi examen. Si es mayor que 1 y menor que 3 el gato me tiene que decir: desempeño bajo. Si está entre 3 y 3.9 el gato me tiene que decir: desempeño básico. Si está entre 4 y 4.5, el gato me tiene que decir: desempeño Alto. Si está entre 4.6 y 5.0, el gato me tiene que decir: desempeño Superior. Cualquier otra nota, No Aplica.