



FUNDAMENTOS DE PROGRAMACIÓN COMPUTACIONAL

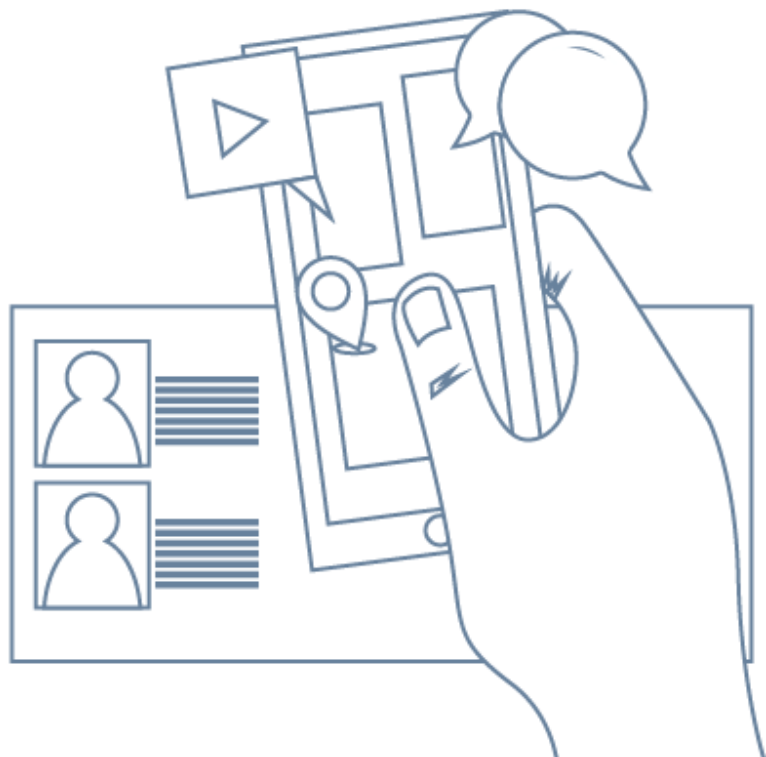
ALGORITMOS Y PROGRAMACIÓN
EN PSEUDOLENGUAJE





FUNDAMENTOS DE PROGRAMACIÓN COMPUTACIONAL

ALGORITMOS Y PROGRAMACIÓN EN PESUDOLENGUAJE





ESCUELA DE CONSTRUCCIÓN E INGENIERÍA

Director de Escuela / Marcelo Lucero

ELABORACIÓN

Experto disciplinar / Marcelo Trigo

Diseñador instruccional / Camila Velásquez

VALIDACIÓN

Experto disciplinar / Nombre experto

Jefa de diseño instruccional y multimedia / Alejandra San Juan Reyes


DISEÑO DOCUMENTO

Equipo de Diseño Instruccional AIEP



Contenido

APRENDIZAJE ESPERADO DE LA SEMANA:	6
1. SCRATCH:	6
1.1. Definición de la herramienta Scratch:	6
1.2. ¿Cómo descargar Scratch?:.....	8
1.3. Componentes del Software de Programación: SCRATCH.....	8
1.4. Scratch y las competencias básicas:	9
1.4.1. Competencia en comunicación lingüística:	9
1.4.2. Tratamiento de la información y competencia:	9
1.4.3. <i>Identificación de problemas, formulación de hipótesis y solución:</i>	10
1.4.4. <i>Desarrollo de las capacidades creativas y la curiosidad intelectual:</i>	10
1.4.5. <i>Competencia social y ciudadana:</i>	10
1.4.6. Competencia para aprender a aprender:	10
1.5. Tipos de Bloques en Scratch:	12
1.5.1. Bloques de Movimiento:.....	12
1.6. Conceptos específicos de programación:	20
1.6.1. Secuencia:	21
1.6.2. Iteración:.....	21
1.6.3. Condicionales:.....	21
1.6.4. Variables:	22
1.6.5. Entrada por teclado:	22
1.6.6. Lógica Booleana:	22
1.6.7. Diseño de interfaz de usuario:	23
1.7. Proposición de trabajo:.....	23
1.7.1. Educación básica:.....	23
1.7.2. Educación Media:.....	24
1.8. Ejemplos Scratch:.....	25
1.8.1. Ejemplo 1(Persona bailando):.....	25
1.8.2. EJEMPLO 2 (Juego Murciélago Abeja):	36
1.8.3. EJEMPLO 3 (SUMAR DOS NUMEROS):	40
1.8.4. EJEMPLO 4 (CONDICIONALES):	46
1.8.5. EJEMPLO 5 Decir cuál es el mayor de dos números:	46
1.8.6. EJEMPLO 5: Decir cuál es el mayor de dos números:	47



1.8.7.	EJEMPLO 6:	47
1.8.8.	EJEMPLO 7:	48
1.8.9.	EJEMPLO 8 (CICLOS):	49
1.8.10.	EJEMPLO 9 (CICLOS):	49
Video:		50
Profundización bibliográfica:		50
IDEAS CLAVE:		51
LINKS/MATERIAL MULTIMEDIA		53
CONCLUSIONES		54
BIBLIOGRAFÍA		55

APRENDIZAJE ESPERADO DE LA SEMANA:

Elaboran algoritmos de programación considerando Scratch como lenguaje de programación visual:

1.12.-Identifica herramienta Scratch, considerando elementos, acciones y comportamientos.

1.13.-Elabora algoritmo de programación considerando funciones básicas de Scratch

1.14.-Relaciona elementos para creación de variables y asignación en el programa, considerando uso de herramienta Scratch

1.15.-Construye programa de animación visual básico, considerando movimientos, apariencia y sonido, según requerimientos de usuarios y uso de herramienta Scratch.

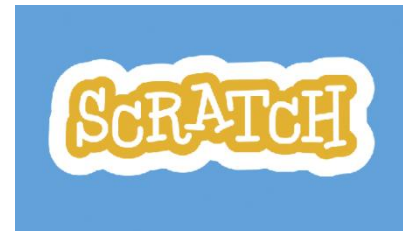
1.16.- Tec-Detecta las causas que originan problemas de acuerdo a parámetros establecidos y en

1. SCRATCH:

1.1. Definición de la herramienta Scratch:

Scratch es un nuevo entorno de programación visual y multimedia basado en Squeak destinado a la realización y difusión de secuencias animadas con o sin sonido y al aprendizaje de programación.

Desarrollado por Media Laboratorios de MIT y por el grupo KIDS de UCLA, Universidad de California, Los Ángeles.



Ofrece una interface intuitiva y muy fácil de comprender, donde podremos manipular imágenes, fotos, sonido, música, etc. y sobre todo una forma de programación visual con todos estos elementos. Podremos crear nuestros personajes, cambiar su apariencia y hacerlos interactuar con los otros. Por último, podremos publicar nuestras creaciones para que todo el mundo pueda verlas.

Con Scratch recuperamos el modelo constructivista de Logo o de los E-Toys de Squeak. Scratch está disponible sobre Windows, Mac OS X y Linux y se difunde bajo licencia libre MIT. Cuando se trabaja con Scratch se comprenden fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa, como son:

- ✓ Los procesos interactivos (bucles),
- ✓ Los criterios condicionales (si, entonces, si-no),
- ✓ Las coordenadas en un plano,
- ✓ Las variables, entre otros.



Además, estos conceptos se aprenden dentro de un contexto significativo y motivador. No es lo mismo comprender el significado de las variables en un contexto de aprendizaje de la programación tradicional, que cuando se utilizan para el control de la visualización de una animación o en juego que uno mismo está construyendo.

Los estudiantes trabajan sus propios proyectos, aprenden mucho sobre el proceso de su concepción. Partiendo de una idea tiene que crear el prototipo funcional, es decir, un modelo, y aportar y experimentar las soluciones que considere. Cuando no funcionan tendrá que volver hacia atrás y corregir. Se crea una espiral continua: partiendo de una idea se crea un proyecto, que da lugar a nuevas ideas, que generan nuevos proyectos, y de este modo indefinidamente.

Estos procesos de concepción de proyectos y su construcción, desarrollan las competencias necesarias para llegar a tener un pensamiento creativo, una comunicación clara, un análisis sistemático, capacidad de colaboración, la reflexión interactiva, entre otros.

De la misma forma que es necesario aprender no solamente a leer sino también a escribir, lo mismo se debe conseguir con la informática, los estudiantes no solamente deben aprender a utilizar e interactuar con el ordenador, también debe aprender a crear con esta herramienta.

Lógicamente no todos van a ser programadores profesionales, pero aprender a programar permite experimentar de forma creativa y ayuda al desarrollo del pensamiento lógico a la vez que comprenden el funcionamiento de las nuevas tecnologías que encontraran en su vida cotidiana.



Figura 1 Proceso de Diseño de Scratch

Fuente: ugelandahuaylas



1.2. ¿Cómo descargar Scratch?:

El Programa de desarrollo de SCRATCH se descarga de los siguientes sitios web:

- » De la página web de Scratch: <http://scratch.mit.edu/>
- » Zona de descarga de Scratch: <http://lk.media.mit.edu/projects/scratch/download/>

1.3. Componentes del Software de Programación: SCRATCH

- » **ESCENARIO:** La zona más grande, blanca, donde está la mascota de Scratch, único actor en ese momento. Ahí se desarrollará la acción.
- » **BOTONES NUEVOS SPRITES:** Justo debajo del escenario hay tres botones que nos van a permitir buscar o incluso crear nuevos actores para nuestra acción.
- » **MODOS PRESENTACIÓN:** Si pinchamos en el botón que queda a la izquierda de los anteriores podemos ampliar el escenario a pantalla completa.
- » **LISTA DE SPRITES:** En la zona que queda debajo de la anterior aparecerán las miniaturas de los sprites que vayan a actuar. Hacer clic para seleccionar y editar un sprite.
- » **BARRA DE HERRAMIENTAS:** Aparece encima del escenario y nos sirven para mover los objetos, copiar, cortar, aumentar su tamaño o disminuir su tamaño.
- » **BANDERA VERDE:** Sirve para ejecutar los guiones o programas que hayamos creado.
- » **BOTON ROJO:** Sirve para detener la acción.
- » **NOTAS DEL PROYECTO:** Para añadir comentarios al proyecto.
- » **PALETA DE BLOQUES:** A la izquierda. Es nuestra caja de instrucciones para que nuestros actores hagan cosas. Son los bloques para crear los scripts o programas. Tenemos 8 cajas diferentes de piezas, para distintas acciones, cada una de un color. Tendremos que ir familiarizándonos con ellas: Movimiento, apariencia, sonido, lápiz, control, sensores, números y variables.
- » **AREA DE SCRIPTS:** Entre las dos zonas anteriores queda otra muy importante, es área de scripts. Aquí iremos arrastrando los bloques de nuestras cajas de piezas para ir construyendo los guiones o programas.
- » **PESTAÑAS:** Justo encima del área de scripts tenemos tres pestañas que nos permiten añadir guiones, disfraces o sonidos al guion que construimos.
- » **ESTILO DE ROTACIÓN:** Nos permite determinar la rotación de los sprites.
- » **INFORMACIÓN SPRITE SELECCIONADO:** En el cuadro aparecerá el nombre del sprite y una miniatura del mismo.

El entorno del programa se divide en las siguientes secciones: **Inicio**

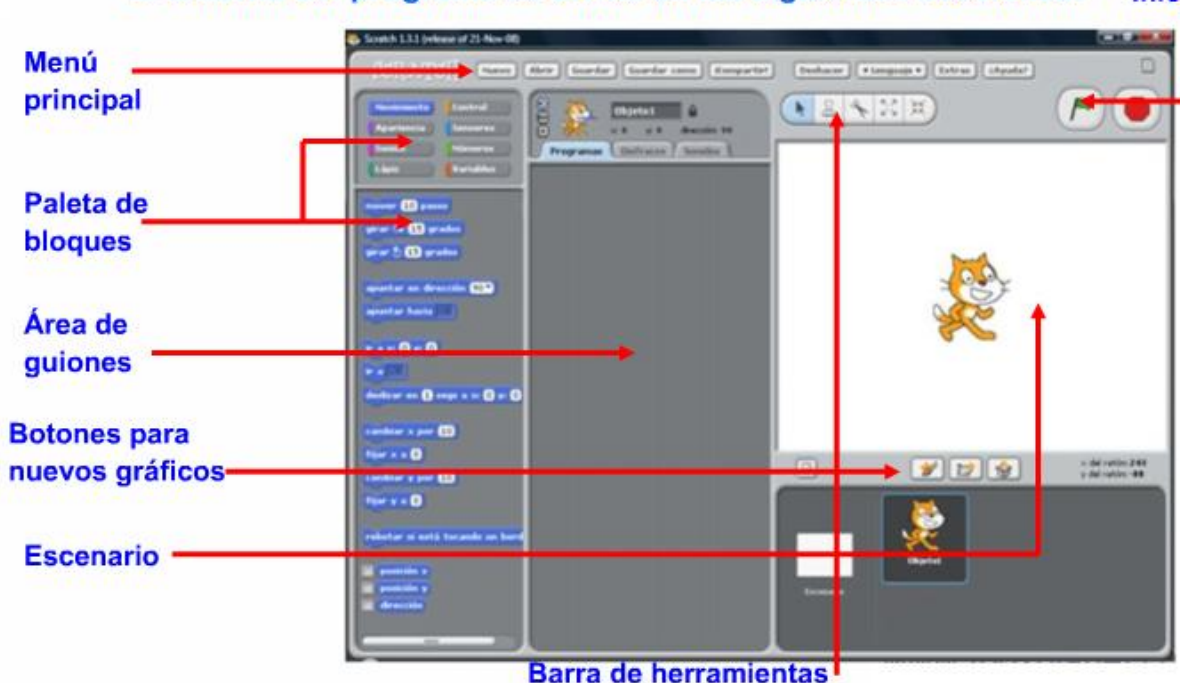


Figura 2 Entorno de Desarrollo de SCRATCH

Fuente: <http://aprendiendioscratch.blogspot.com/2015/03/estructura-y-herramientas.html>



A continuación, te dejamos un pequeño video tutorial sobre Scratch: La programación divertida:

<https://www.youtube.com/watch?v=6JyBCx6WGJQ&feature=youtu.be>

1.4. Scratch y las competencias básicas:

1.4.1. Competencia en comunicación lingüística:

Una comunicación efectiva hoy en día requiere de más habilidades que simplemente leer y escribir. Con Scratch, los jóvenes aprenden a ser capaces de manipular e integrar diversos tipos de información para conseguir expresarse de forma creativa y persuasiva.

1.4.2. Tratamiento de la información y competencia:

Trabajar con Scratch permitirá a los estudiantes aprender a seleccionar, crear y manejar información de diversos tipos: texto, imágenes, secuencias animadas y sonido. Al tiempo que los estudiantes adquieren experiencia trabajando con esta información, se vuelven cada vez más receptivos y críticos analizando la información que les llega del mundo que les rodea.



1.4.3. Identificación de problemas, formulación de hipótesis y solución:

Scratch permite aprender a través de un contexto significativo basado en el proceso de diseño. Crear un proyecto con Scratch requiere pensar una idea, dividir esa idea en pasos e implementar esos pasos mediante el sistema de programación de bloques del programa. Está diseñado para poder ver el resultado de la programación en el acto, por lo que los estudiantes aprenden este proceso de manera interactiva.

1.4.4. Desarrollo de las capacidades creativas y la curiosidad intelectual:

Scratch estimula el pensamiento creativo, una habilidad muy valorada hoy en día. Scratch estimula a los estudiantes para que busquen soluciones innovadoras a problemas inesperados que surgen durante el proceso de diseño.

1.4.5. Competencia social y ciudadana:

Al ser una herramienta que permite compartir proyectos muy fácilmente, puede ser utilizada para incentivar el debate entre los jóvenes de cuestiones con importancia social, no solo en el entorno educativo, sino que se puede llevar a un nivel de discusión internacional, gracias a la Comunidad Scratch.

1.4.6. Competencia para aprender a aprender:

Aprendiendo a programar con Scratch, los jóvenes van descubriendo el razonamiento crítico y el pensamiento sistemático. En sus proyectos necesitan coordinar el tiempo y las interacciones entre diferentes personajes, y su habilidad para programar esto, les proporciona una experiencia directamente relacionada con la detección de problemas, la crítica constructiva, el ensayo-error, etc. conceptos importantes dentro del pensamiento sistemático. Al trabajar en proyectos que son significativos para los jóvenes, sus propias ideas les proporcionan la motivación adecuada para sobrellevar las dificultades y retos que les plantea el proceso de diseño.

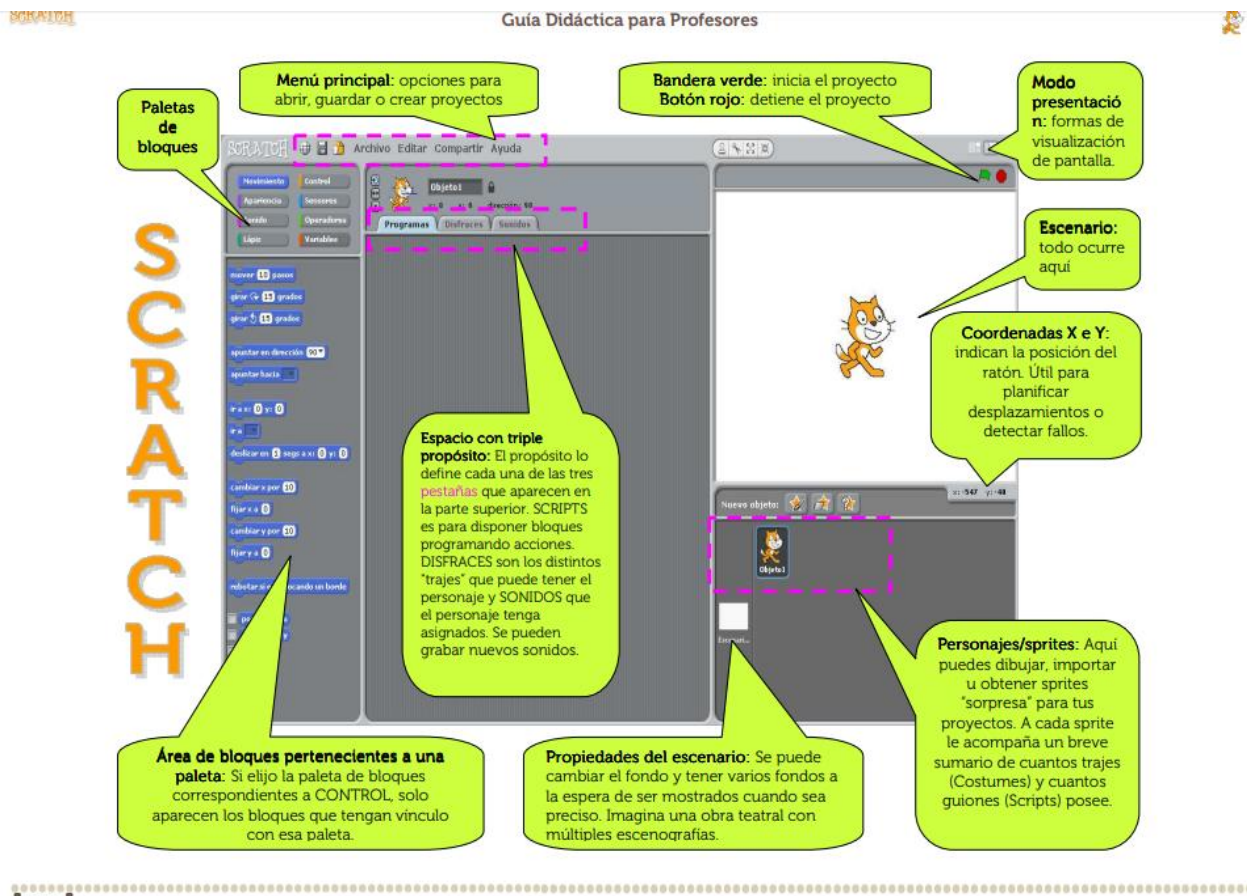


Figura 3 Entorno de Desarrollo de SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.5. Tipos de Bloques en Scratch:

1.5.1. Bloques de Movimiento:

A continuación, se declaran los distintos tipos de movimientos que realiza SCRATCH:

MOVIMIENTO	
	Mueve el Objeto hacia adelante o hacia atrás.
	Rota el Objeto en el sentido de las manecillas del reloj.
	Rota el Objeto en el sentido contrario a las manecillas del reloj.
	Apunta el Objeto en la dirección especificada (0=arriba; 90=derecha; 180=abajo; -90=izquierda).
	Apunta el Objeto hacia el puntero del ratón o hacia otro Objeto.
	Mueve el Objeto hacia una posición específica de X, Y en el escenario.
	Mueve el Objeto a la ubicación del puntero del ratón o de otro Objeto.
	Mueve el Objeto suavemente a una posición determinada en un lapso de tiempo específico.
	Cambia la posición X del Objeto en una cantidad determinada (incrementa).
	Fija la posición X del Objeto a un valor específico.
	Modifica la posición Y del Objeto en una cantidad determinada (incrementa).
	Fija la posición Y del Objeto a un valor específico.
	Gira el Objeto en sentido contrario, cuando este toca un borde del escenario.
	Informa la posición X del Objeto. (Rango entre -240 a 240)
	Informa la posición Y del Objeto (Rango entre -180 a 180)
	Informa la dirección del Objeto (0=arriba; 90=derecha; -90=izquierda; 180=abajo)

Figura 4 Bloques de movimiento de SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.5.2. Bloques de Apariencia:

A continuación, se declaran los distintos tipos de apariencia que realiza SCRATCH

APARIENCIA	
	Modifica la apariencia del Objeto cambiando de disfraz.
	Cambia el disfraz del Objeto por el siguiente disfraz en la lista de disfraces (cuando llega al final del listado de estos, vuelve a comenzar con el primer disfraz).
	Informa el número correspondiente al presente disfraz del Objeto
	Modifica la apariencia del escenario cambiando a un fondo diferente.
	Modifica la apariencia del escenario pasando al siguiente fondo disponible en el listado de estos.
	Reporta el número del fondo actual del escenario.
	Despliega una nube de diálogo del Objeto durante un lapso de tiempo determinado
	Despliega una nube de diálogo del Objeto (se puede eliminar esta burbuja de diálogo ejecutando este bloque sin texto alguno).
	Despliega una nube de pensamiento del Objeto durante un determinado lapso de tiempo.
	Despliega una nube de pensamiento del Objeto.
	Modifica un efecto visual del Objeto en una cantidad especificada (use el menú desplegable para seleccionar el efecto).
	Establece un efecto visual a un número dado (la mayoría de efectos visuales se ubica en un rango de 0 a 100).
	Limpia o borra todos los efectos gráficos de un Objeto
	Modifica el tamaño del Objeto en una cantidad especificada (incrementa).
	Ajusta el tamaño del Objeto en un porcentaje (%) específico respecto a su tamaño original.
	Informa el tamaño del Objeto como porcentaje (%) de su tamaño original.

Figura 5 Bloques de apariencia de Scratch

Fuente: www.paginaspersonales.deusto.es

1.5.3. Bloques de SONIDO

A continuación, se declaran los distintos tipos de sonido que realiza SCRATCH:

SONIDO	
	Comienza la reproducción del sonido seleccionado del menú desplegable, e inmediatamente pasa al siguiente bloque aunque el sonido se esté ejecutando aún.
	Reproduce un sonido y espera hasta que el sonido termine, antes de continuar con el bloque siguiente.
	Detiene todos los sonidos.
	Reproduce un determinado número de sonido de tambor, seleccionado del menú desplegable, durante un número específico de pulsos.
	Reproduce una nota musical (número altos para tonos altos) durante un número específico de pulsos.
	Descansa, no toca nada, durante un número específico de pulsos.
	Establece el tipo de instrumento que usa el Objeto para los bloques de tocar notas (cada Objeto tiene su propio instrumento).
	Modifica el volumen del sonido del Objeto en un valor especificado (incrementa o decrementa el volumen).
	Fija el volumen del sonido del Objeto a un valor específico.
	Informa el volumen del sonido del Objeto.
	Modifica el <i>tempo</i> del Objeto en una cantidad específica (incremental).
	Fija el tempo del Objeto a un valor especificado de pulsos por minuto.
	Informa el tempo del Objeto en pulsos por minuto.

Figura 6 Bloques de sonido SCRATCH
Fuente: www.paginaspersonales.deusto.es

1.5.4. Bloques de LÁPIZ:

A continuación, se declaran los distintos tipos de lápiz que realiza SCRATCH








LÁPIZ	
borrar	Borra todas las marcas de lápiz y de sellos (estampados) del Escenario.
bajar lápiz	Baja el lápiz del Objeto, de manera que este pinte a medida que se mueve.
subir lápiz	Levanta el lápiz del Objeto, de manera que no pinte cuando se mueva.
fijar color de lápiz a 	Establece el color del lápiz, basado en la selección hecha en la paleta de color.
cambiar color del lápiz por  10	Modifica el color del lápiz en una cantidad específica.
fijar color de lápiz a  0	Establece el color del lápiz a un valor determinado. (color-lápiz=0 en el borde rojo del arco iris; color-lápiz=100 en el borde azul del arco iris)
cambiar intensidad de lápiz por  10	Modifica la intensidad del lápiz en una cantidad especificada (incrementa).
fijar intensidad de lápiz a  50	Establece un valor específico para la intensidad del lápiz (sombra-lápiz=0 es muy oscura; sombra-lápiz=100 es muy clara).
cambiar tamaño de lápiz por  1	Cambia el grosor del lápiz (incrementa en una cantidad específica).
fijar tamaño de lápiz a  1	Establece el grosor del lápiz.
sellar	Estampa o copia la imagen del Objeto en el Escenario.

Figura 7 Bloques de lápiz de SCRATCH
Fuente: www.paginaspersonales.deusto.es

1.5.5. Bloques de CONTROL:

A continuación, se declaran los distintos tipos de Control que realiza SCRATCH:












CONTROL	
	Ejecuta el programa que tiene debajo al hacer clic en la bandera verde.
	Ejecuta el programa que tiene debajo al presionar una tecla específica.
	Ejecuta el programa que tiene debajo al hacer clic en un Objeto.
	Espera un número determinado de segundos y continúa luego con el bloque siguiente.
	Ejecuta continuamente los bloques en su interior.
	Ejecuta, un número específico de veces, los bloques en su interior.
	Envía un mensaje a todos los Objetos, activándolos para que hagan algo y espera a que todos terminen antes de continuar con el siguiente bloque.
	Envía un mensaje a todos los Objetos y luego continúa con el bloque siguiente sin esperar a que se realicen las acciones de los Objetos activados.
	Ejecuta el programa que tiene debajo cuando recibe un mensaje específico "enviar a todos" (Broadcast).
	Comprueba continuamente si una condición es verdadera; cada que es verdadera, ejecuta los bloques en su interior.
	Si la condición es verdadera, ejecuta los bloques en su interior.

Figura 8 Bloques de control de SCRATCH
Fuente: www.paginaspersonales.deusto.es

1.5.6. Bloques de SENSORES:

A continuación, se declaran los distintos tipos de Sensores que realiza SCRATCH:

y del ratón	Informa la posición "Y" del puntero del ratón.
¿ratón presionado?	Informa verdadero, si el botón del ratón está presionado.
¿tecla espacio presionada?	Informa verdadero, si una tecla específica está presionada.
¿tocando ?	Informa verdadero, si el Objeto está tocando un Objeto específico, un borde o el puntero del ratón (seleccionados del menú desplegable).
¿tocando el color ?	Informa verdadero, si el Objeto está tocando un color específico. (Haga clic en la paleta de color y luego utilice el gotero para seleccionar el color).
¿color sobre ?	Reporta verdadero si el primer color (dentro del Objeto), está tocando un segundo color (tanto en el fondo como en otro Objeto). Haga clic en la paleta de color y luego utilice el gotero para seleccionar el color.
distancia a	Informa la distancia desde un Objeto específico o desde el puntero del ratón.
reiniciar cronómetro	Fija el cronómetro en 0.
cronómetro	Reporta el valor del cronometro en segundos (el cronómetro siempre está contando).
posición x de Objeto1	Informa una propiedad o variable de otro Objeto.
volumen del sonido	Reporta el volumen de los sonidos captados por el micrófono del computador (entre 1 y 100).
¿sonido fuerte?	Reporta verdadero, si el volumen del sonido captado por el micrófono del computador es mayor de 30 (en escala de 1 a 100).
valor del sensor deslizador	Informa el valor de un sensor específico. Para usar este bloque se necesita tener un "PicoBoard" conectado a su computador. Para aprender más consulte: http://www.playfulinvention.com/picoboard.html
¿sensor botón presionado activado?	Informa verdadero, si un sensor específico está presionado. Para usar este bloque se necesita tener un "PicoBoard" conectado a su computador. Para aprender más consulte:

Figura 9 Bloques de sensores de scratch
Fuente: www.paginaspersonales.deusto.es

1.5.7. Números:























NÚMEROS	
	Suma dos números.
	Resta dos números
	Multiplca dos números.
	Divide dos números.
	Selecciona un número entero al azar dentro de un rango especificado.
	Informa verdadero, si el primer valor es menor que el segundo.
	Reporta verdadero, si dos valores son iguales.
	Informa verdadero, si el primer valor es mayor que el segundo.
	Informa verdadero, si ambas condiciones son verdaderas.
	Informa verdadero, si una de las dos condiciones es verdadera.
	Reporta verdadero, si la condición es falsa; reporta falso si la condición es verdadera.
	Reporta el resultado de una función seleccionada (abs, raíz cuadrada, sin, cos, tan, asin, acos, atan, ln, log, e [^] , 10 [^]) aplicada a un número específico.
	Informa el resto (módulo) de la división del primer número por el segundo número.
	Informa el entero más cercano a un número.

Figura 10 NUMEROS DE SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.5.8. Variables:

VARIABLES	
	Permite crear y nombrar una nueva variable. Cuando usted genera una variable, aparecen los bloques correspondientes a ella. Se puede escoger si la variable es para todos los Objetos (global) o solo para un Objeto (local)
	Borra todos los bloques asociados con una variable
	Informa el valor de la variable
	Modifica la variable en una cantidad determinada (Si se tiene más de una variable, utilice el menú desplegable para seleccionar el nombre de la variable)
	Fija la variable a un valor específico.
	Muestra el monitor de la variable en el escenario
	Esconde el monitor de la variable para que no aparezca en el escenario
	Permite crear y nombrar una nueva lista. Cuando se genera una lista, aparecen los bloques para esa lista. Se puede escoger si la lista es para todos los Objetos (global) o solo para un Objeto (local)

	Borra los bloques asociados a una lista.
	Reporta todos los elementos que tiene la lista.
	Añade el elemento especificado al final de la lista (el elemento puede ser un número o una cadena de letras u otros caracteres).
	Borra uno o todos los elementos de una lista. Se puede escoger del menú desplegable o usar un número para indicar qué elemento borrar. Si escoge "último" borrará el último elemento de la lista. Si escoge "todos" borra todo lo que contiene la lista. Borrar, decrementa la longitud de la lista.
	Inserta un elemento en un lugar específico de la lista. Se puede escoger del menú desplegable o usar un número para indicar dónde insertar el elemento dentro de la lista. Si escoge "último" adiciona el elemento al final de la lista. Si se escoge "cualquiera" lo inserta aleatoriamente en la lista. La longitud de la lista se incrementa en 1.
	Reemplaza un elemento de la lista con un valor específico. Se puede escoger del menú desplegable o usar un número para especificar el elemento que va a reemplazar. Si escoge "último", reemplaza el último elemento de la lista. Si escoge "cualquiera" reemplaza aleatoriamente un elemento de la lista. La longitud de la lista no se modifica.
	Reporta el elemento en una ubicación específica dentro de la lista. Se puede insertar este bloque dentro de otros bloques como
	Reporta cuántos elementos hay en la lista.

Figura 11 VARIABLES DE SCRATCH
Fuente: www.paginaspersonales.deusto.es

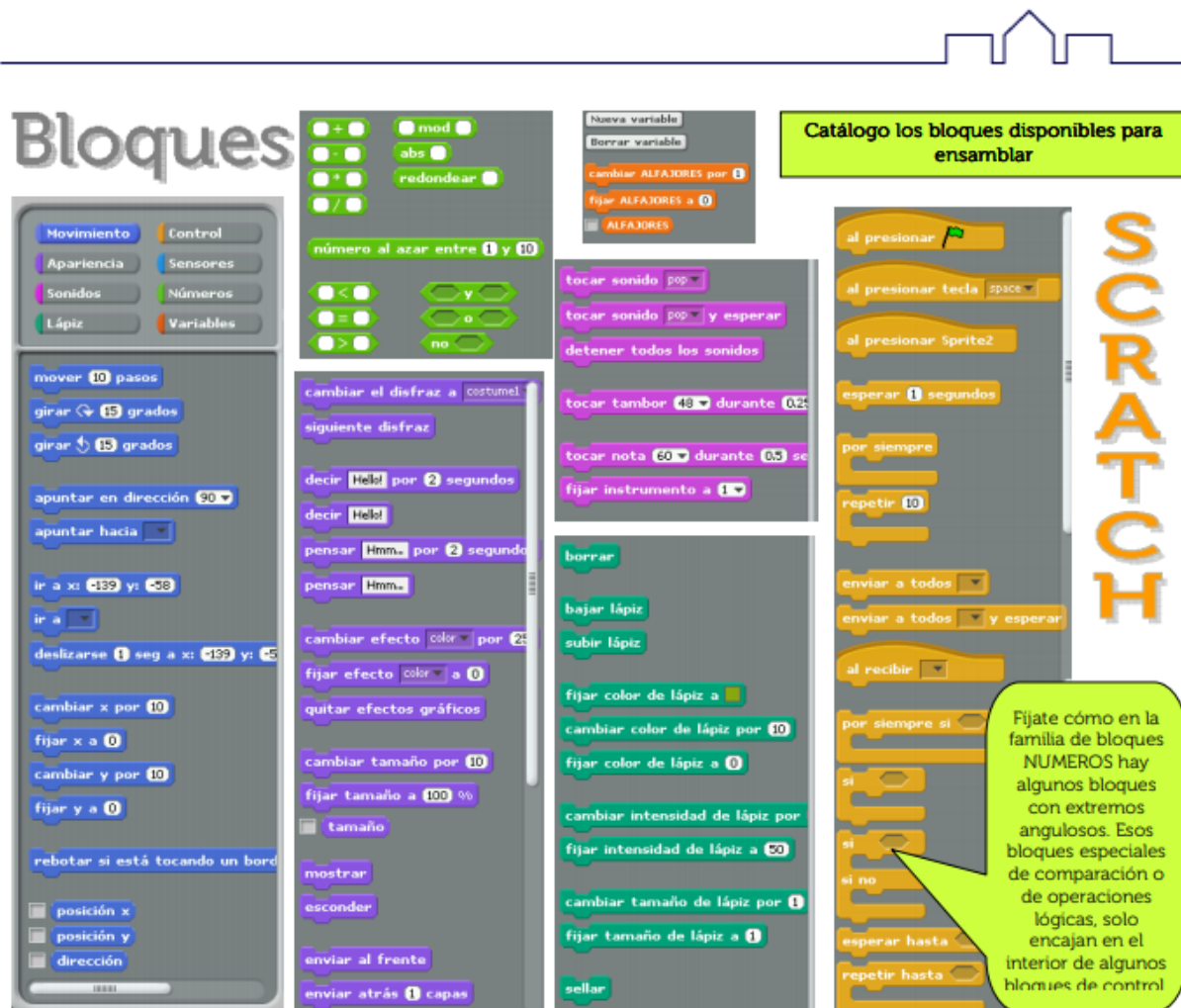


Figura 12 RESUMEN DE BLOQUES
Fuente: www.paginaspersonales.deusto.es

1.6. Conceptos específicos de programación:

En esta Sección se explicará una serie de conceptos específicos de programación, con el objetivo de que los profesores se familiaricen con ellos. En ningún caso se pretende que los alumnos se aprendan estos términos de memoria, sino que mediante el uso del programa vayan descubriendo cómo funciona y para qué sirven cada una de las partes que lo componen:

1.6.1. Secuencia:

La secuencia en programación estructurada es la línea de pasos que se siguen para aplicar un desarrollo. En Scratch se ve de la siguiente forma:


CONCEPTO	EXPLICACIÓN	EJEMPLO
Secuencia	Para crear un programa en <i>Scratch</i> , se necesita pensar sistemáticamente el orden de los pasos a seguir.	

Figura 13 SECUENCIA En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.2. Iteración:

La iteración o bien llamada ciclos, recordar que es la repetición de varios bloques de programación:

Iteración (looping)	Los bloques 'Por siempre' y 'Repetir' pueden ser usados para la iteración (repetir una serie de instrucciones).	
----------------------------	---	--

Figura 14 ITERACIÓN En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.3. Condicionales:

Los condicionales, son comparaciones que tienen una respuesta, verdadero o falso y dependiendo de la respuesta, ejecutan acciones determinadas.

Condicionales	Los bloques 'Si...' y 'Si...no...' dan cuenta de una condición.	
----------------------	---	--

Figura 15 CONDICIONALES En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.4. Variables:

Las variables son posiciones de memoria que guardan valores determinados, por lo general ingresados por un usuario.


Variables	Los bloques de 'Variable' permiten crear variables y usarlas en un programa. Las variables pueden almacenar números o <i>strings</i> (cadenas de caracteres). <i>Scratch</i> respalda variables tanto globales como específicas a un objeto.	
------------------	--	--

Figura 16 VARIABLES En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.5. Entrada por teclado:

El ingreso de datos por teclado es importante en la comunicación del programa con el usuario.


Entrada de teclado	El bloque 'Preguntar y esperar' invita al usuario a tipear en el teclado. 'Respuesta' almacena la entrada tecleada.	
---------------------------	---	--

Figura 17 ENTRADAS DE TECLADO En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.6. Lógica Booleana:

La lógica es parte de la programación, la mayoría de las estructuras están relacionadas con algún resultado lógico.


Lógica Booleana	Los bloques '...y...', '...o...', 'no...' son ejemplos de lógica booleana.	
------------------------	--	--

Figura 18 LOGICA BOOLEANA En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.6.7. Diseño de interfaz de usuario:

La comunicación con el usuario no solo se produce en los datos que esta entrega sino también en la conexión grafica que entrega nuestro programa.

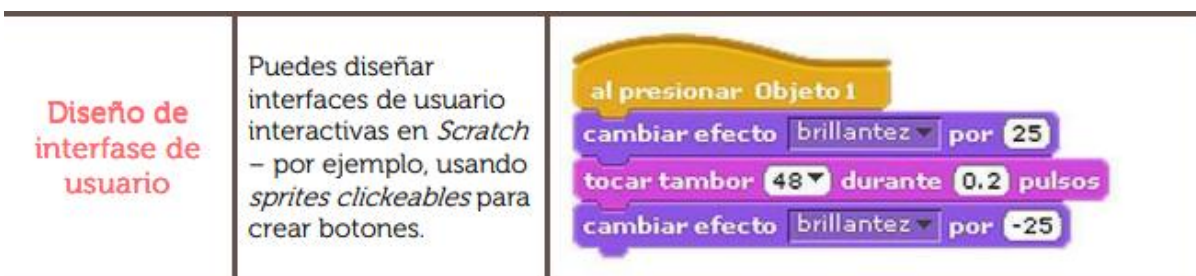


Figura 19 DISEÑO INTERFAZ DE USUARIO En SCRATCH

Fuente: www.paginaspersonales.deusto.es

1.7. Proposición de trabajo:

Una de las cosas más interesantes de Scratch es la capacidad de utilizar materiales capacidad de utilizar materiales externos a la propia herramienta, elaborados por él, elaborados por el alumnado, elaborados por el alumnado, de manera que lo que alumnado ellos preparan (sus propias fotografías, sus dibujos e imágenes, su música...) puede ser incorporado al programa a través de las opciones de opciones de importación opciones de importación. Importación.

Otra de las características que convierten a Scratch en una herramienta colaborativa herramienta colaborativa interesante es la posibilidad de posibilidad de posibilidad de compartir las creaciones compartir las creaciones compartir las creaciones que desarrollamos con la herramienta en la misma Web de la aplicación, a través de la opción "Compartir", disponible desde el propio entorno de desarrollo. Para esto hace falta haber creado previamente un perfil (una cuenta de usuario) en la web en la web oficial de Scratch.

El propio web programa se encarga de comprimir todo el material utilizado y enviarlo a la web. También se puede fomentar entre el alumnado entre el alumnado entre el alumnado el compartir recursos, sin necesidad de mandarlos a la web. Así todos puedan utilizar los materiales elaborados por todos, y así crear trabajos derivados crear trabajos derivados crear trabajos derivados de los mismos.

1.7.1. Educación básica:

Scratch se puede comenzar a utilizar con alumnado a partir de los 8 años de edad, aunque para comenzar a utilizar algunas de las funcionalidades se debería esperar, hasta los 10 años de edad más o menos (3º de Primaria).

Esta herramienta permite que se haga un uso de ella con diferentes niveles de diferentes niveles de complejidad. Se puede comenzar a utilizar la herramienta cuando se desee, incluso con alumnado muy joven, haciendo uso únicamente de proyectos ya realizados, los proyectos ya realizados cuales pueden visualizar o visualizar o visualizar o



interactuar interactuar interactuar, para ir conociendo la herramienta, familiarizarse con el entorno y los personajes, etc.

A partir del Segundo Ciclo de Educación Primaria, la propuesta sería comenzar a utilizar la herramienta como una forma de contar hi utilizar la herramienta como una forma de contar historias como una forma de contar historias. Se pueden ubicar Historias Y personajes y crear diálogos entre ellos, hacer que se desplacen por la pantalla, etc.

También se utilizar, como otras herramientas de programación, como una calculadora. Y en cualquiera de los cursos de Primaria se puede comenzar a trabajar con el análisis de pequeños programas ya realizados análisis de pequeños programas ya realizados para mejorarlos, cambiar los elementos de una aplicación para personalizarlos, etc. En el siguiente link encontrareis ejemplos de recursos didácticos creados con Scratch, por los alumnos de magisterio de la Universidad del País Vasco. La mayoría de ellos están en Euskara. <http://scratch.mit.edu/tags/view/iktmagisteritza>

1.7.2. Educación Media:

En esta etapa se pueden trabajar más profundamente los conceptos abstractos.

Se puede comenzar a asignar eventos a los personajes asignar eventos a los personajes asignar eventos a los personajes, a utilizar el teclado y el lado y el ratón para manejar los objetos, y sobre todo a asignar ratón para manejar los objetos asignar acciones que ocurren cuando varios objetos interaccionan entre sí. se pueden conseguir efectos interesantes, que se irán complicando sucesivamente según se vayan adquiriendo nuevas habilidades.

En esta etapa, se puede empezar a trabajar con los “mensajes” que unos objetos pueden enviar a otros. Cuando se produce una condición determinada, un objeto “envía un mensaje” a otros objetos, de manera que estos lleven a cabo una acción determinada.

1.8. Ejemplos Scratch:

1.8.1. Ejemplo 1(Persona bailando):

PASO 1: Abrimos el programa. Una vez abierto veremos en pantalla la interfaz gráfica del programa similar a la imagen siguiente.

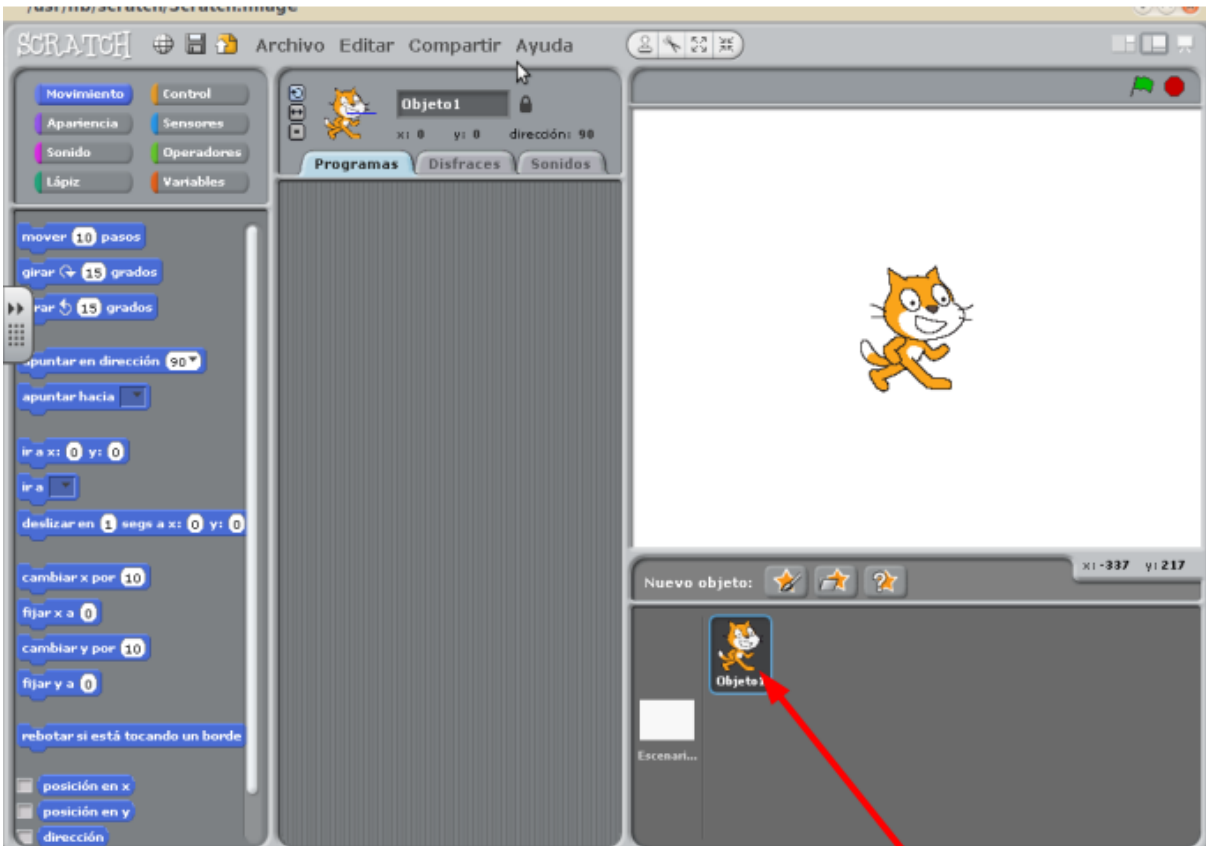


Figura 20 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 2: Por defecto aparece el Objeto1 (gato naranja scratch). Seleccionamos el Objeto1 y lo vamos a borrar haciendo clic con el botón derecho del ratón "Borrar". Tal y como se ve en la siguiente figura:

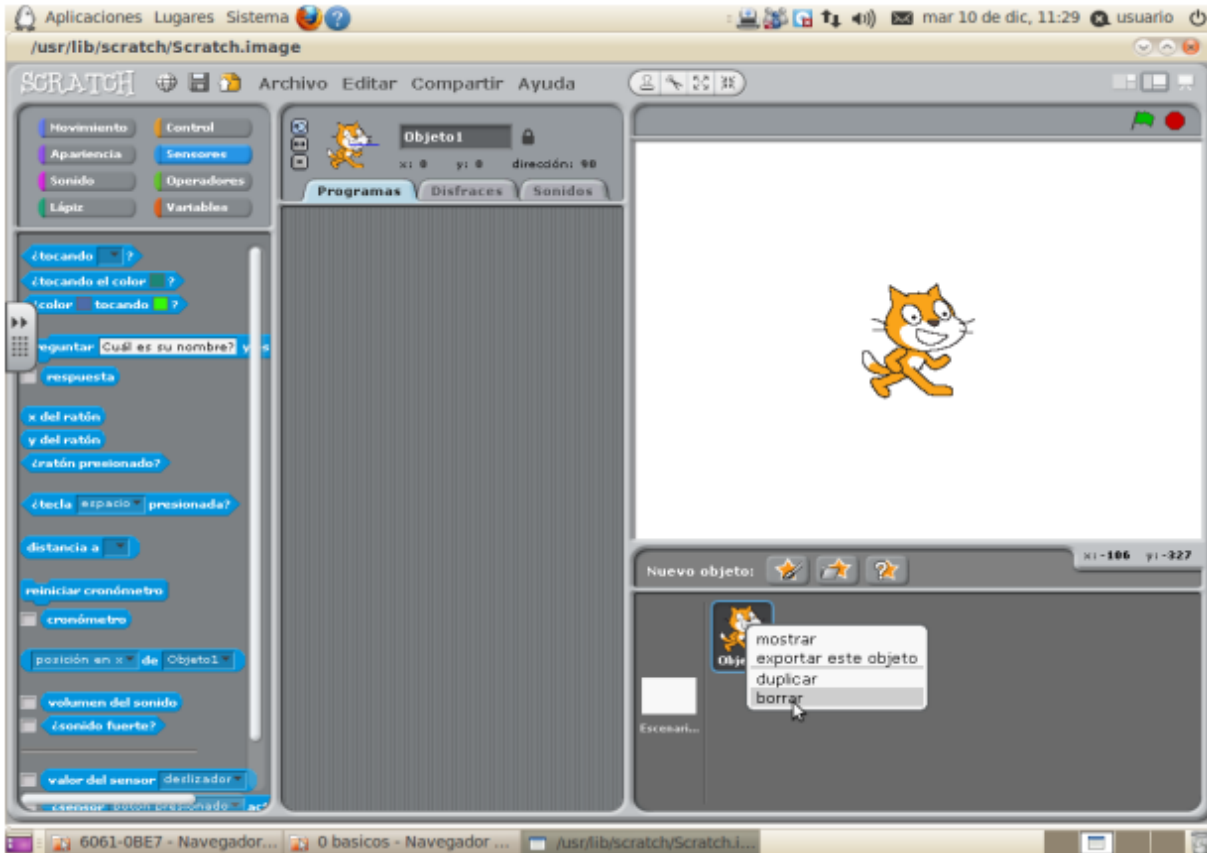


Figura 21 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 3: A continuación, seleccionamos el escenario, en la pestaña Fondos importaremos la imagen "spotlightstage" desde la carpeta Indoors. (Para este ejemplo, descargue una imagen desde Internet y aplique como se le indica)

A continuación seleccionamos el escenario, en la pestaña Fondos importaremos la imagen "spotlightstage" desde la carpeta Indoors.

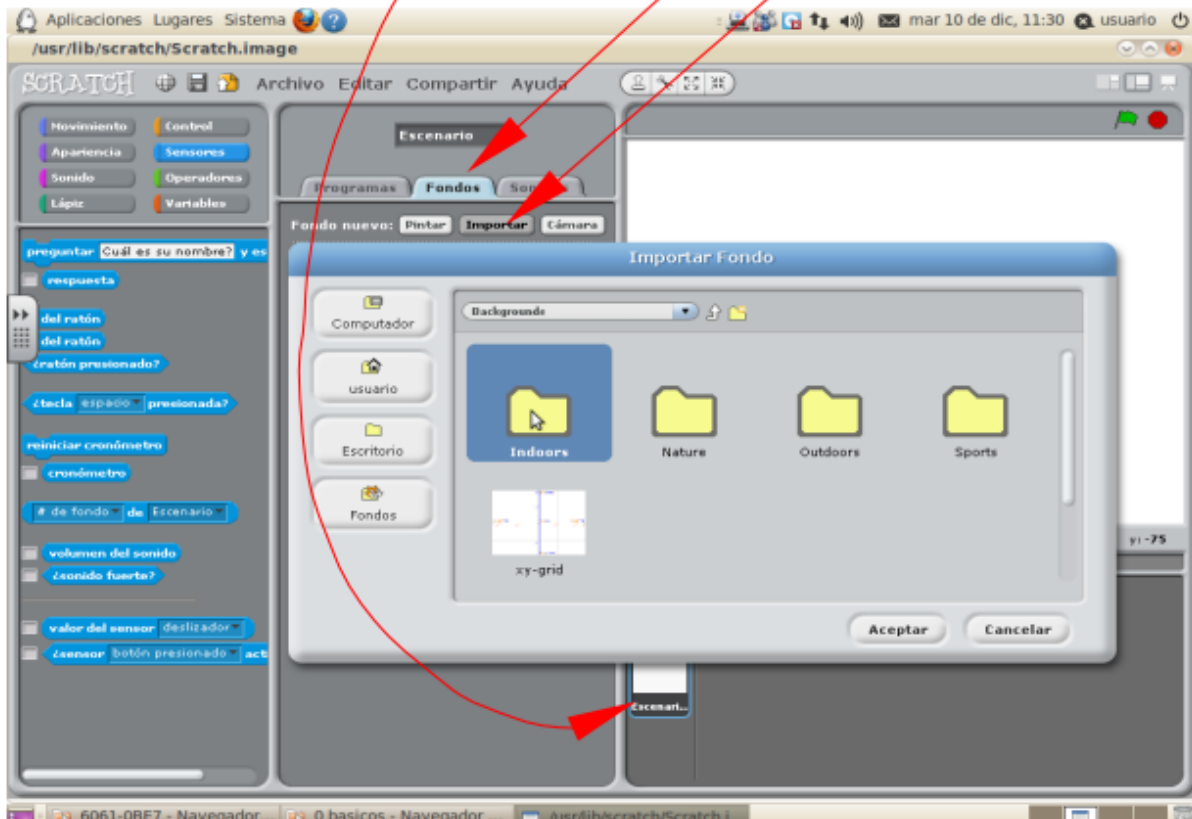


Figura 22 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 4:

4) Borramos el **fondo1** que viene por defecto.

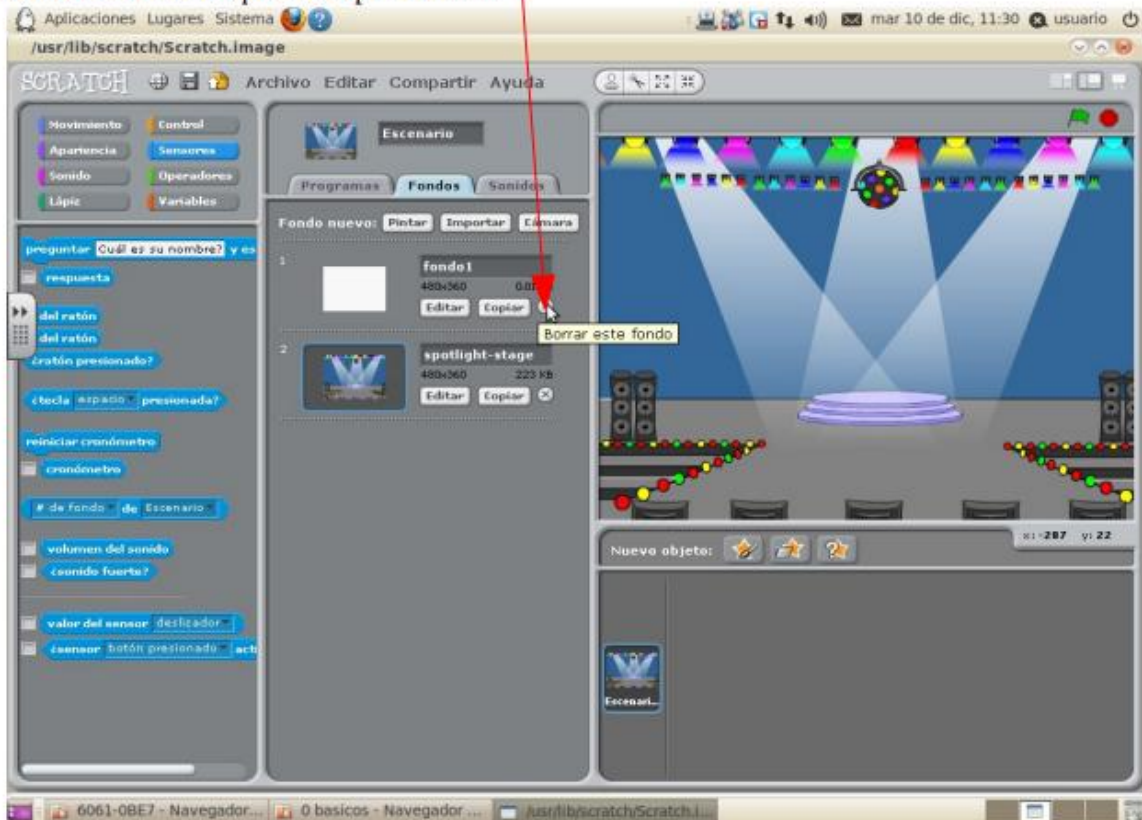


Figura 23 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 5: Ahora añadiremos el personaje, para ello insertamos un nuevo objeto desde la carpeta People, escogemos a "dan1"

Ahora añadiremos el personaje, para ello insertamos un nuevo objeto desde la carpeta **People** escogemos a **"dan1"**.

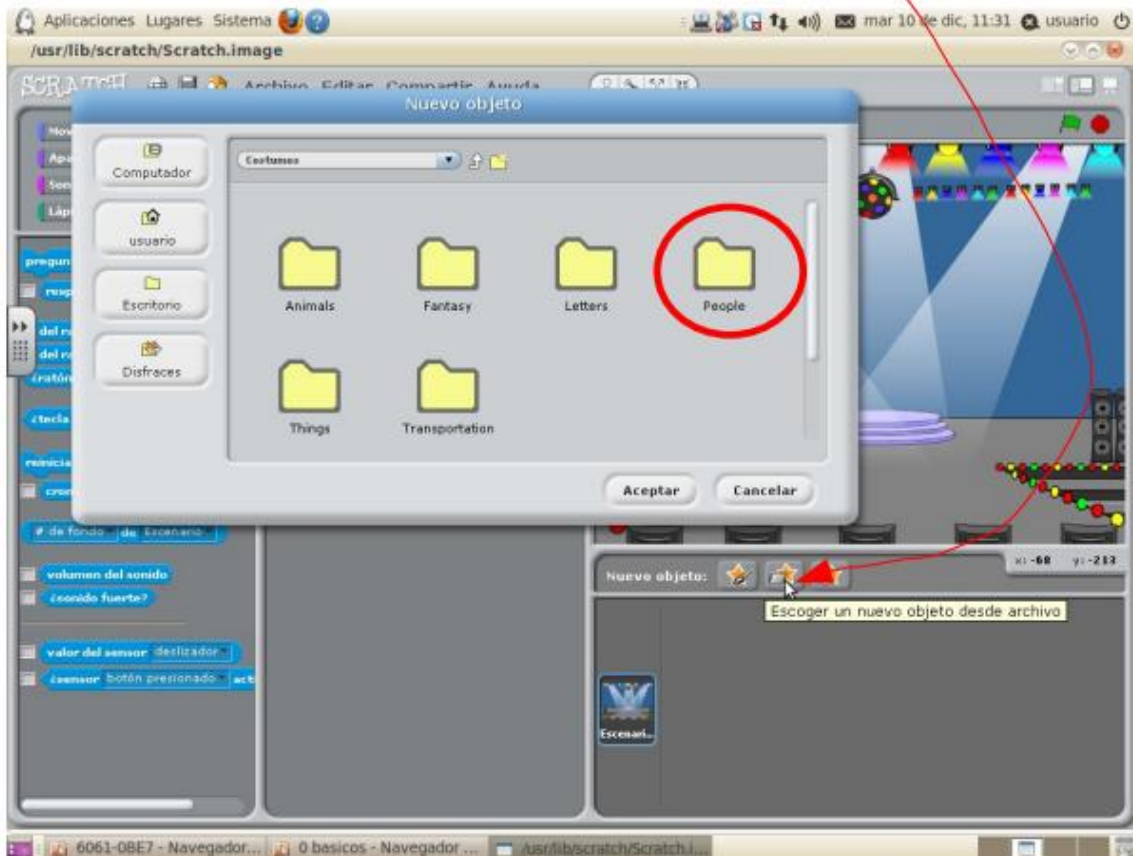


Figura 24 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 6:

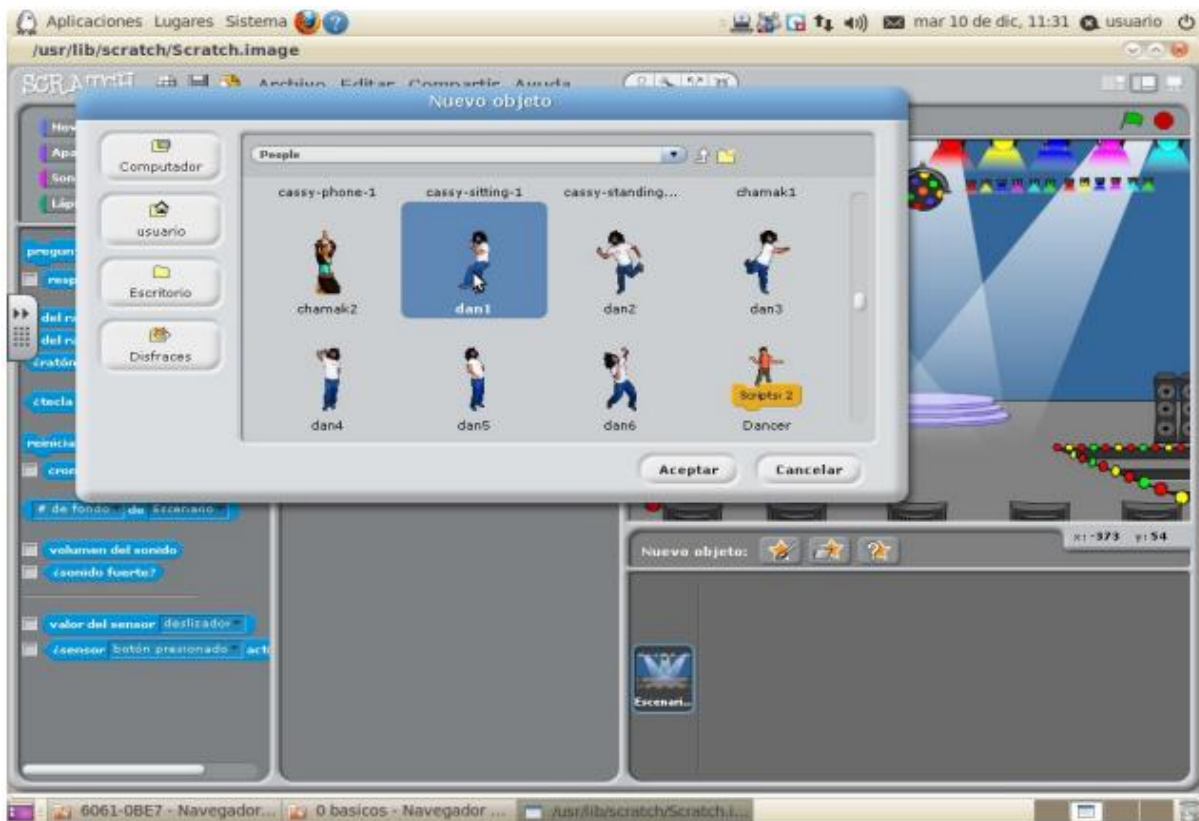


Figura 25 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 7: Teniendo el personaje seleccionado, nos vamos a la pestaña Disfraces e importaremos el resto de los disfraces de dan (dan1, dan2 ...dan6)

- 6) Teniendo el personaje seleccionado, nos vamos a la pestaña **Disfraces** e importaremos el resto de disfraces de **dan** (dan1,dan2,.....dan6).



Figura 26 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 8: En este momento ya tenemos todo lo que necesitamos (escenario y personaje). Ahora hay programaremos lo que queremos que haga el personaje. Seleccionamos el Objeto1 y en la pestaña Programas iremos poniendo los bloques de instrucciones simplemente arrastrándolos desde la izquierda.

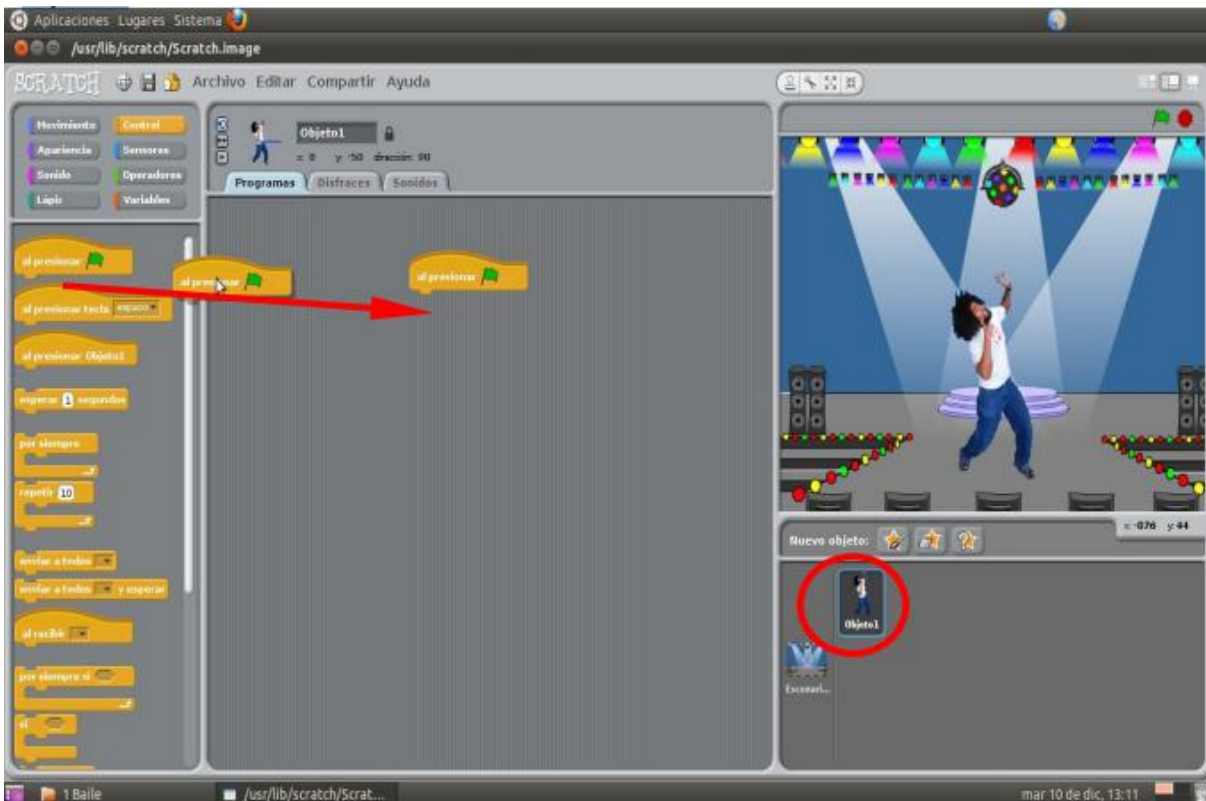


Figura 27 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 9: Este fragmento de código significa que el objeto1 se posicinará en el punto (x=0, y= -50) del escenario al pulsar la bandera verde. (La bandera verde pone en funcionamiento el programa y el círculo rojo lo detiene).

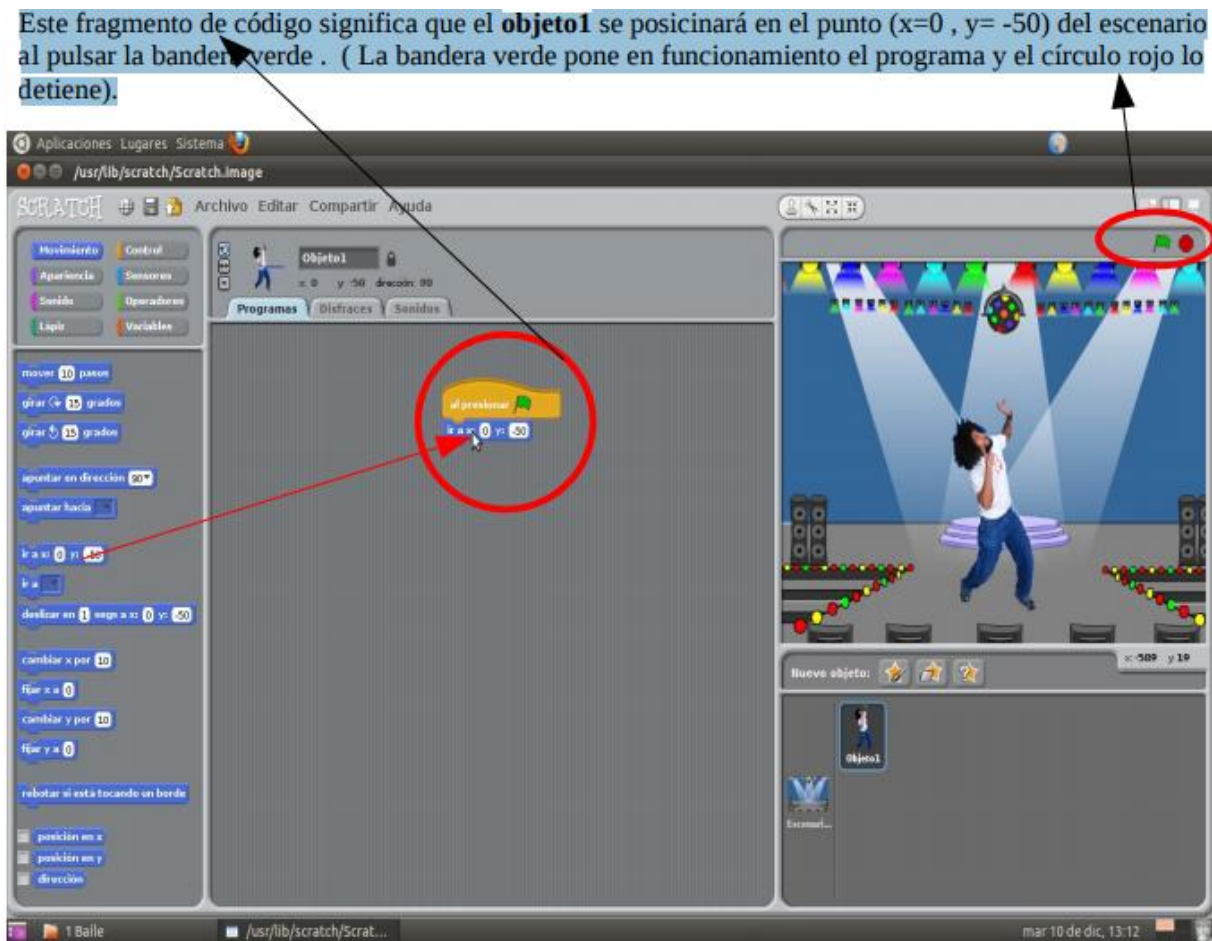


Figura 28 EJEMPLO 1

Fuente: <https://www.danieldon.com/scratch>

PASO 10: Ponemos como disfraz inicial a dan1.

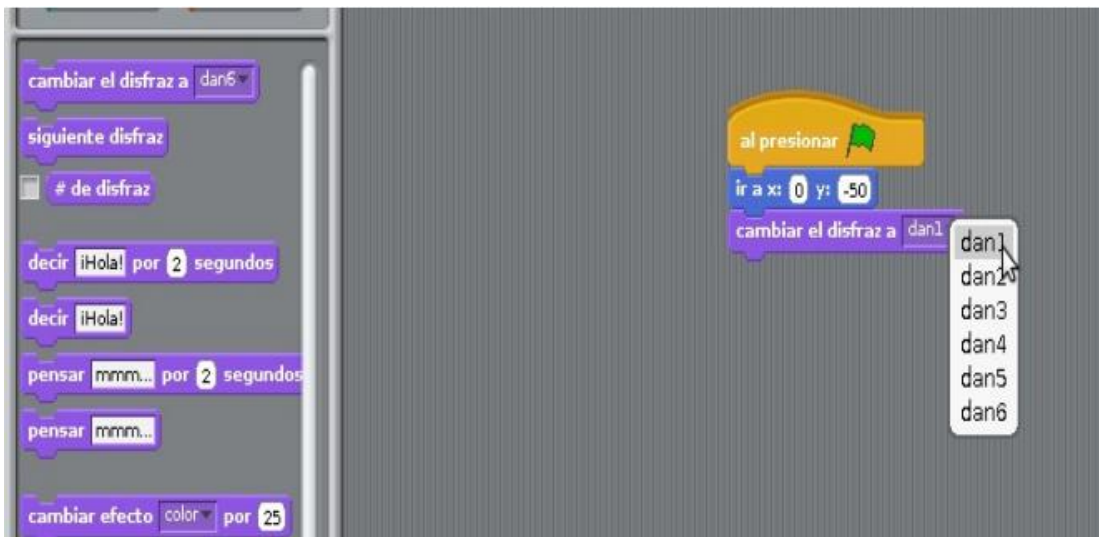


Figura 29 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 11: Añadimos un bucle infinito (bloque por siempre). Dentro de este bucle ponemos un bloque para que el personaje cambie al siguiente disfraz de la lista (dan1, dan2, ...dan6) dándonos la impresión de que el personaje se pone a bailar. Hay que poner un retardo de unos 0,5 segundos para que el cambio de disfraz no sea demasiado rápido.

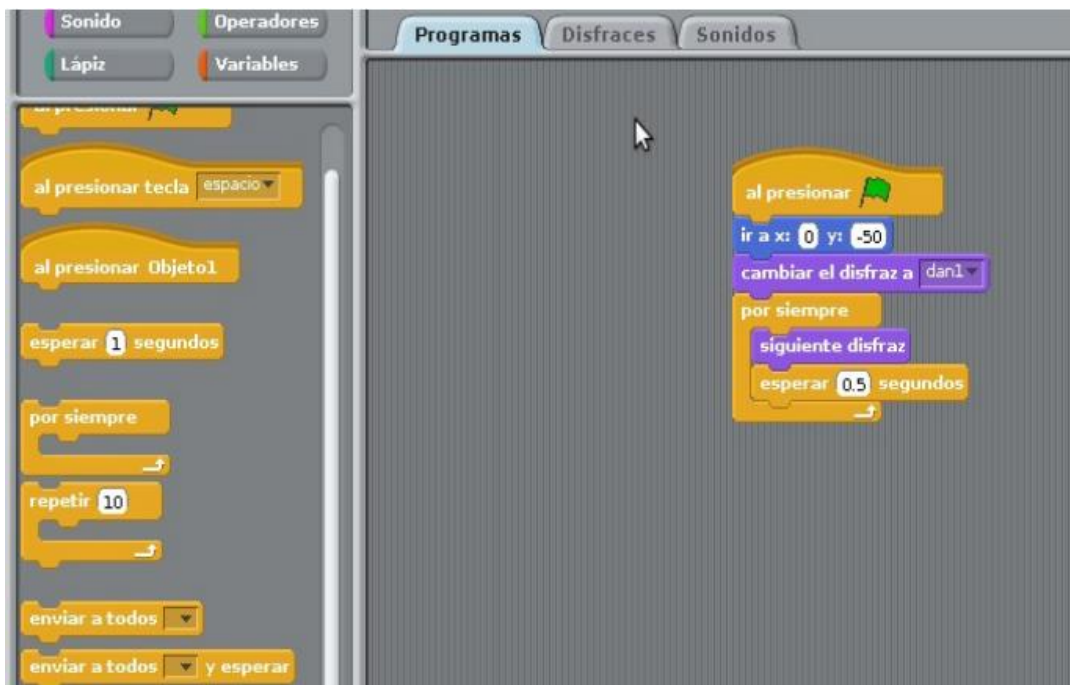


Figura 30 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

PASO 12: Si queremos incorporar cierta interacción desde teclado, para que el personaje se mueva por la pantalla al pulsar ciertas teclas, podemos hacerlo añadiendo el siguiente código.

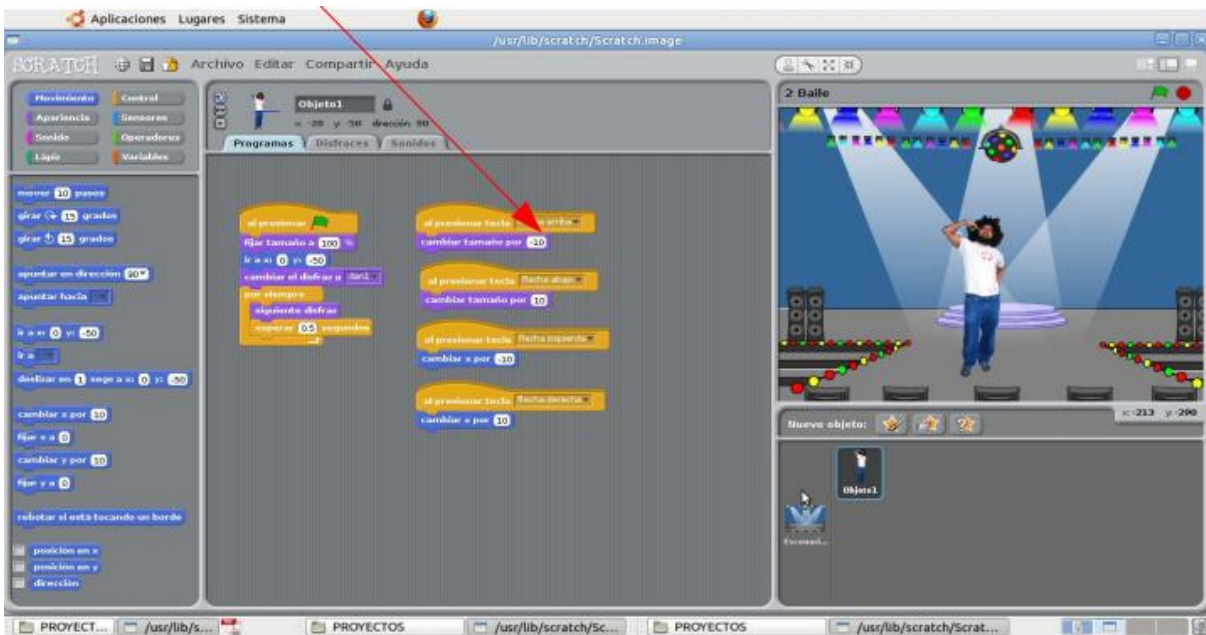


Figura 31 EJEMPLO 1

Fuente: <https://www.danieldona.com/scratch>

O mejor este otro , que hará que la reacción sea más rápida. Aquí se ponen unos bloque condicionales de tipo Sí (El bloque condicional Sí ejecuta las instrucciones que están dentro de él siempre y cuando su prueba lógica sea verdadera).

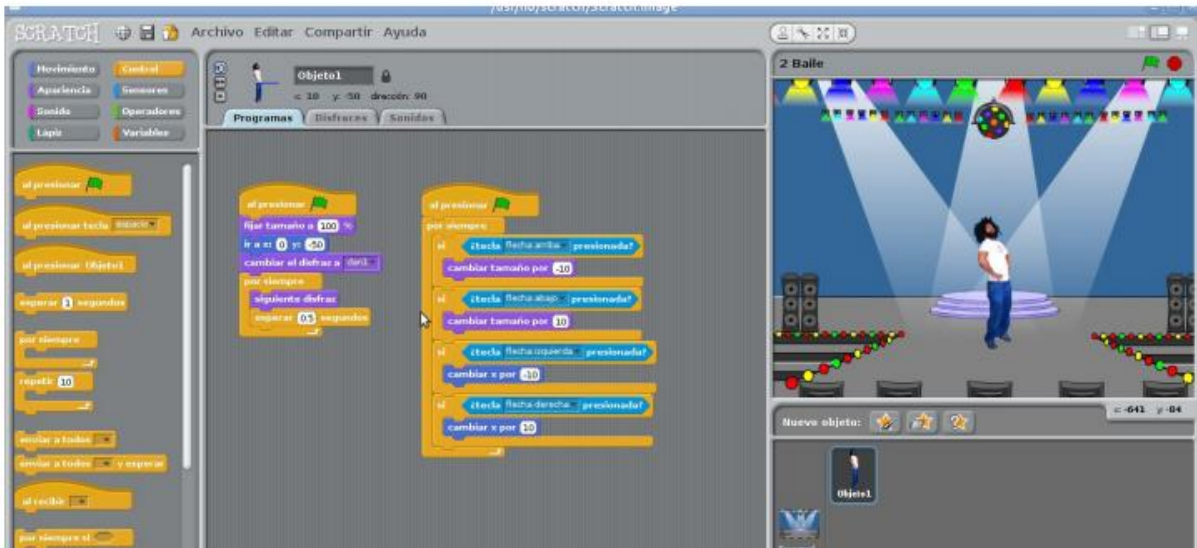


Figura 32 EJEMPLO 1

Fuente: <https://www.danieldon.com/scratch>

PASO FINAL: OBSERVACIÓN cuando guardes el proyecto guardarlo en la carpeta USUARIO o de lo contrario te puede dar un error

1.8.2. EJEMPLO 2 (Juego Murciélago Abeja):

En este proyecto vamos a crear un pequeño juego en el que aparece un murciélago volando que será perseguido por una abeja. El murciélago lo vamos a controlar con las flechas arriba, abajo, izquierda y derecha. El juego acaba cuando la abeja toca al murciélago.

PASO 1: Preparamos el escenario importando el fondo “woods”

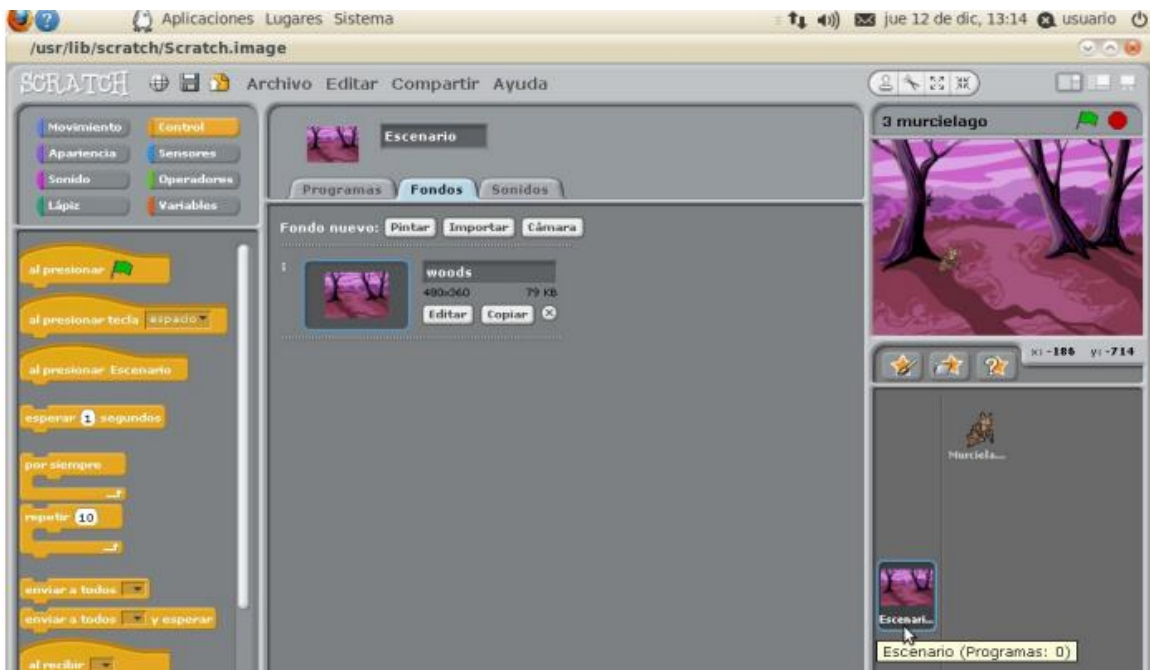


Figura 33 EJEMPLO 2

Fuente: <https://www.danieldona.com/scratch>

PASO 2: Insertamos desde la carpeta Animals el objeto bat1-a. Luego en la pestaña Disfraces importamos bat1- b. Con estos dos disfraces conseguiremos que el murciélago mueva las alas.

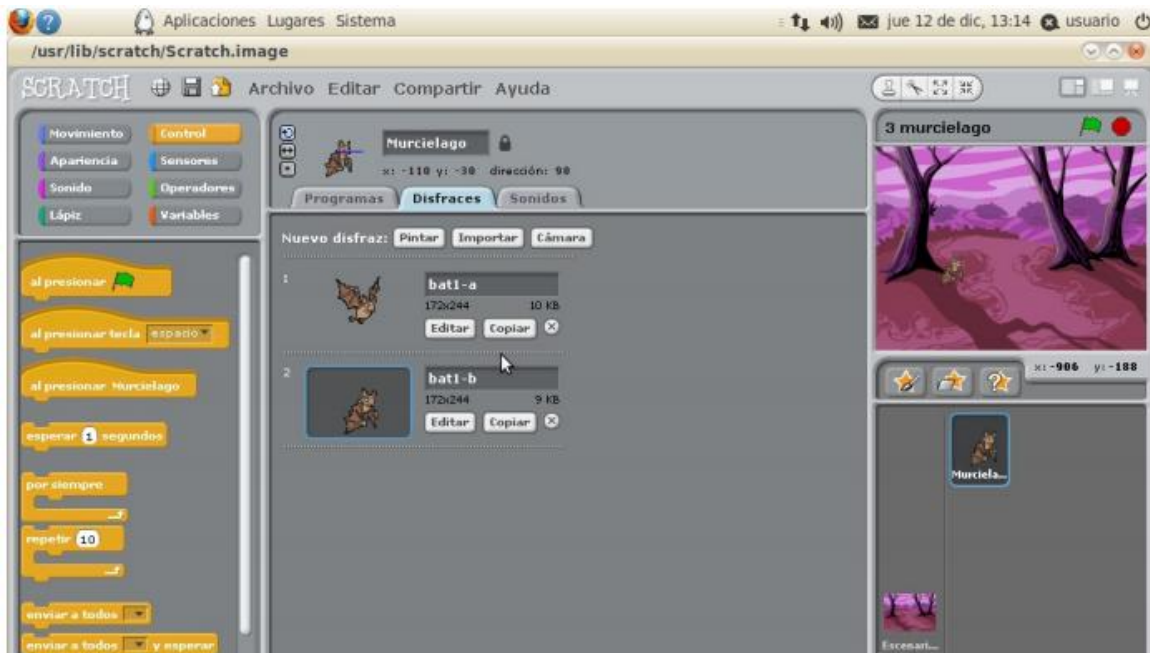


Figura 34 EJEMPLO 2

Fuente: <https://www.danieldona.com/scratch>

PASO 3: La añadimos a el Murciélago el código necesario para que:

- » Aparezca en el punto $x=0$, $y=0$ al comenzar el juego.
- » Tenga un tamaño del 30% para hacerlo un poco más pequeño.
- » El control de las teclas para moverlo (teclas flechas).



Figura 35 EJEMPLO 2

Fuente: <https://www.danieldona.com/scratch>

PASO 4: Ponemos otro bloque para que parezca que mueve las alas. Se consigue cambiando de disfraz.

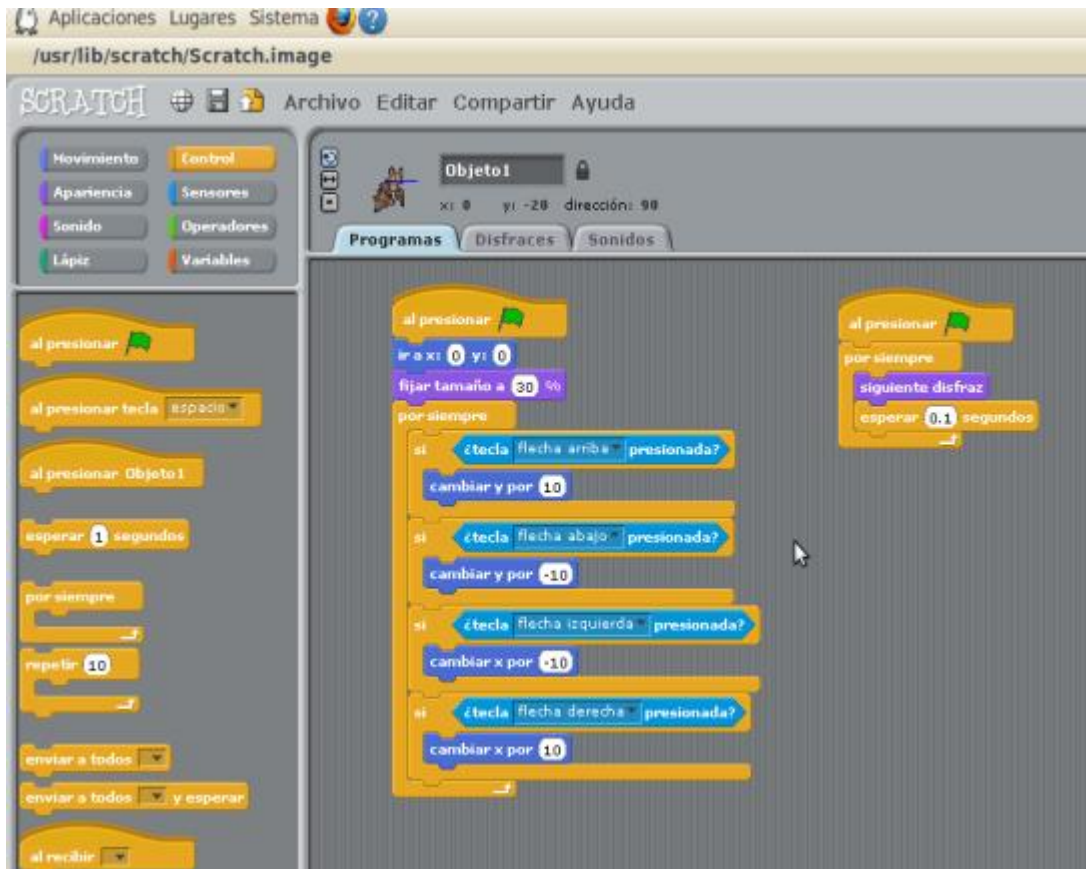


Figura 36 EJEMPLO 2

Fuente: <https://www.danieldona.com/scratch>

PASO 5: Para conseguir que el juego termine cuando la abeja toca a el murciélago completamos el bloque con una instrucción de tipo condicional “si”.

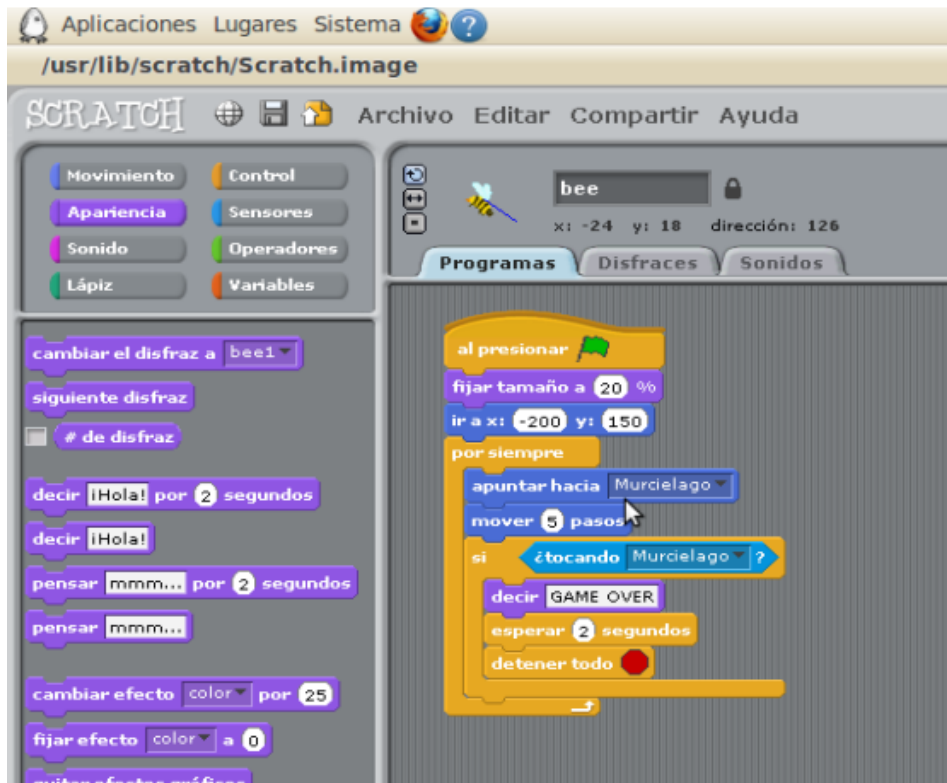


Figura 37 EJEMPLO 2

Fuente: <https://www.danieldona.com/scratch>

1.8.3. EJEMPLO 3 (SUMAR DOS NUMEROS):

PASOS:

1. Crear las variables que permitirán guardar los valores necesarios para hacer la suma
 - a. Clic en la opción Variables
 - b. Clic en Nueva Variable
 - c. Escribimos el nombre de la variable a utilizar, siendo: numero1
 - d. Clic en aceptar
 - e. Repetimos los pasos del b al d para las variables: numero2 y resultado



Figura 38 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

2. Quitamos la selección de las tres variables para que no se vean en el escenario al iniciar la animación.



Figura 39 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

3. Podemos hacer uso del gato o agregamos otro objeto cualquiera, quien nos servirá de intermediario entre la animación y el usuario. utilizaremos el gato.



Figura 40 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

4. Inicializamos la animación con el Comando Al presionar Bandera Verde
 - a. debemos tener seleccionado el gato
 - b. vamos a la opción Control
 - c. Arrastramos el comando Al Presionar la Bandera Verde hacia el centro de la aplicación (Debe quedar seleccionado la pestaña Programas)



Figura 41 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

5. Seleccionamos la opción Variables

- Ubicamos tres veces la opción Fijar después de el comando Al Presionar la Bandera Verde
- Cambiamos los nombres de la Variable de la Pestaña Negra que tiene el comando al lado izquierdo.
- esto se hace con el fin de que las variables siempre comiencen con valor cero (0), para no tener problemas los valores más adelante.
- Ubicamos tres veces la opción Esconder variable, para que las tres variables siempre comiencen encondidas.

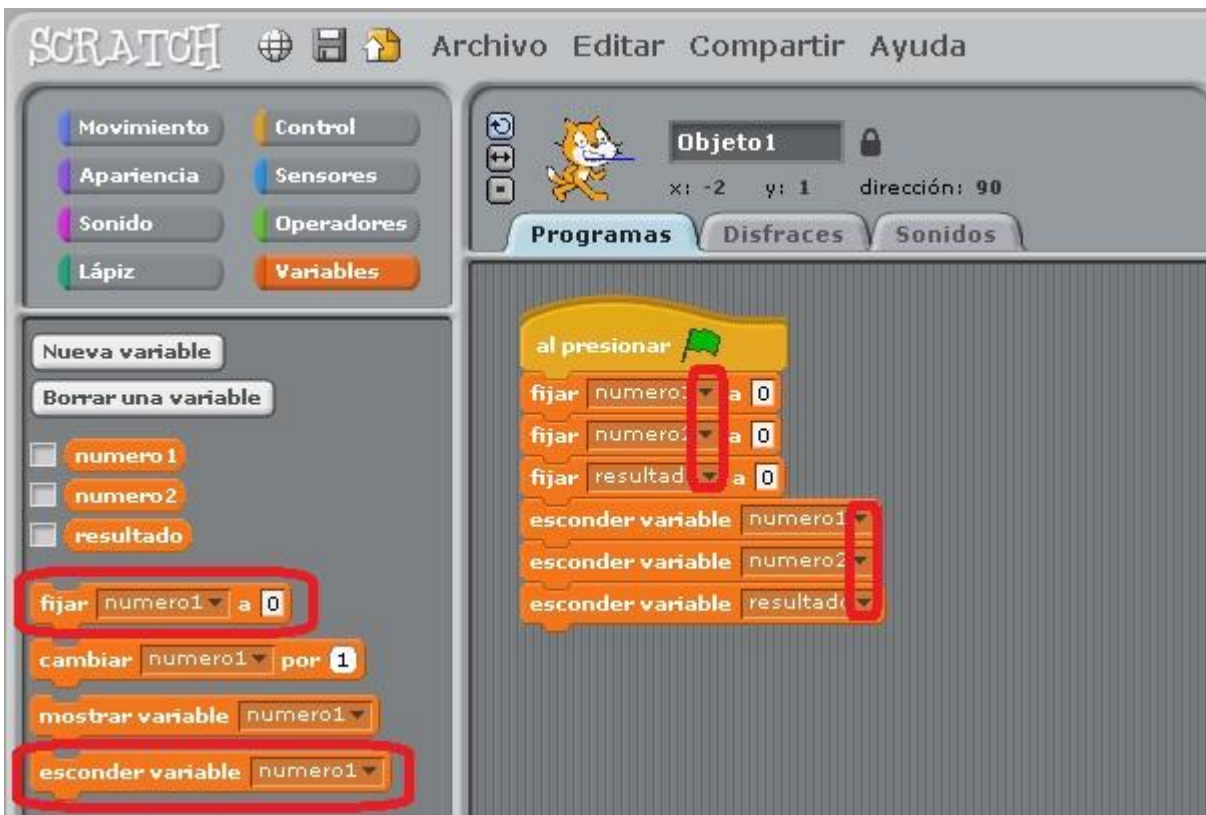


Figura 42 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

6. Utilizando la opción Sensores y Variables solicitaremos los números a sumar y los guardamos en las variables correspondientes
 - a. Agregamos el comando Preguntar (Sensores) y Escribimos una frase que pida el primer número a sumar.
 - b. Agregamos el comando Cambiar (Variables) y agregamos el comando Respuesta (Sensores) en la opción donde está el número 1.
 - c. Hacemos lo mismo para el segundo número.

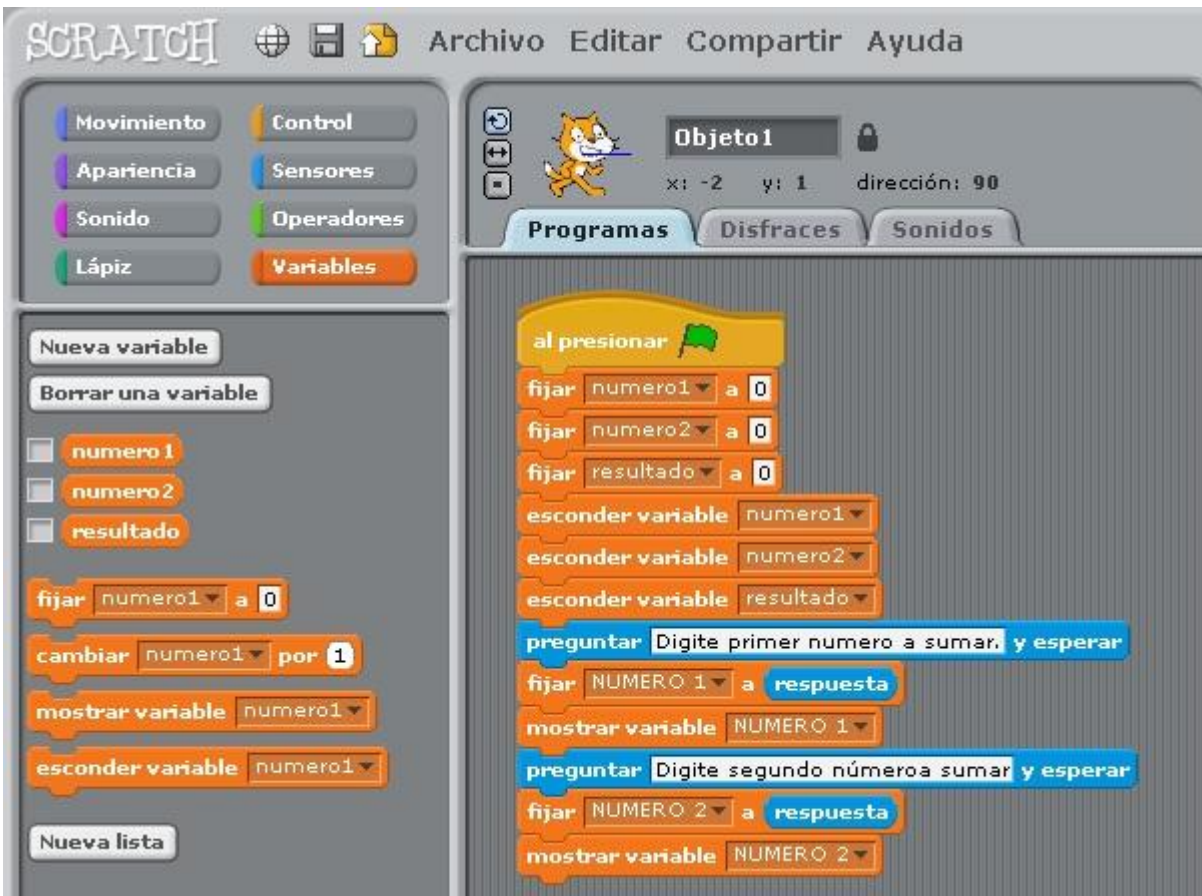


Figura 43 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

7. Realizamos la suma de los dos números digitados

- Utilizamos el comando decir (Apariencia) para la frase: La suma de los dos números arroja como resultado:
- agregamos el comando cambiar (Variables) para la variable **resultado**
- agregamos el comando **suma** (operadores) y ubicamos las dos variables a sumar.
- agregamos el comando **mostrar variable** (variables)



Figura 44 EJEMPLO 3

Fuente: <https://www.danieldona.com/scratch>

1.8.4. EJEMPLO 4 (CONDICIONALES):

Escribe el programa para que poder escribir por pantalla de qué fruta tienes más, si peras o manzanas.

Primero crearemos dos variables: peras y manzanas. Fijamos peras a un valor y manzanas a otro. Desde la pestaña Control seleccionamos nuestras cajas para trabajar con condicionales. El resultado será:



Figura 45 EJEMPLO 4

Fuente: <https://www.danieldona.com/scratch>

1.8.5. EJEMPLO 5 Decir cuál es el mayor de dos números:



Figura 46 EJEMPLO 5

Fuente: <https://www.danieldona.com/scratch>

1.8.6. EJEMPLO 5: Decir cuál es el mayor de dos números:



Figura 47 EJEMPLO 6

Fuente: <https://www.danieldona.com/scratch>

1.8.7. EJEMPLO 6:

Tengo la nota de mi examen. Si es menor que 5 el gato me tiene que decir que estoy suspenso. Si es mayor que 5, tiene que decirme que es notable si está entre 5 y 8, y me tiene que decir sobresaliente si es mayor o igual que 8.



Figura 48 EJEMPLO 6

Fuente: <https://www.danieldona.com/scratch>

1.8.8. EJEMPLO 7:

Tengo 3 peras, 4 plátanos, 5 yogures y 7 flanes. Si en la familia somos 4 o más tomamos lácteo (yogur o flan) y si somos menos fruta (pera o plátano). Siempre tomamos del que tenga más. Escribe el programa en Scratch que nos indique que vamos a tomar.

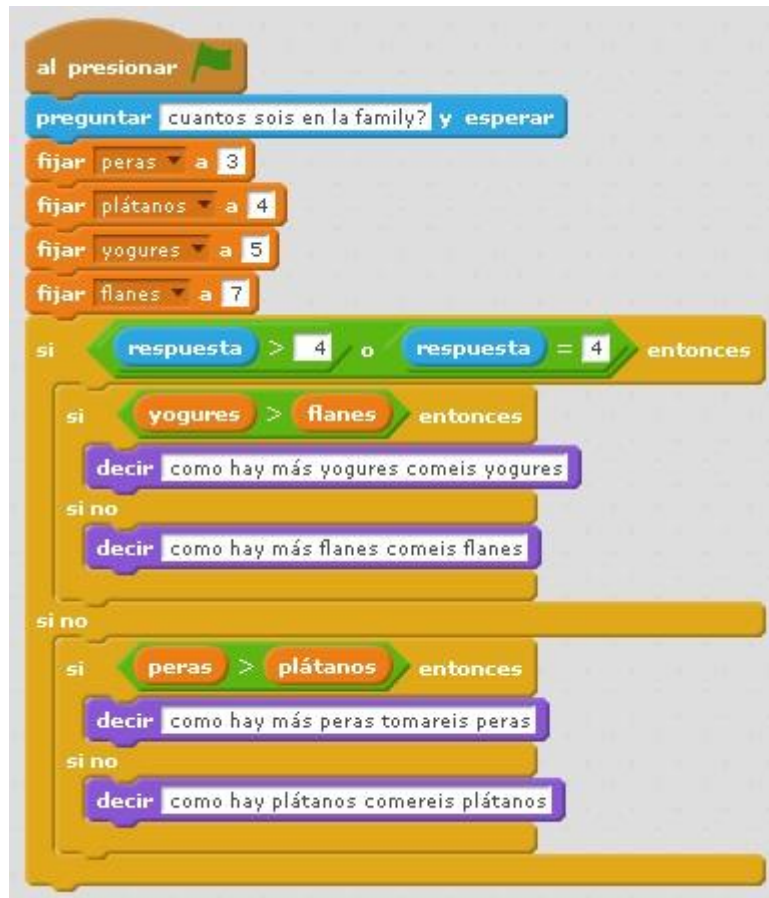


Figura 49 EJEMPLO 7

Fuente: <https://www.danieldona.com/scratch>

1.8.9. EJEMPLO 8 (CICLOS):

Queremos que el gato cada 25 pasos diga "Ya estoy aquí", espere un segundo y lo haga hasta que haya dado 150 pasos.

Para ello tendrá que repetir 6 veces ($150 / 25 = 6$). Usaremos las pestañas Control/ Repetir, Movimiento y Control/ esperar.



Figura 50 EJEMPLO 8

Fuente: <https://www.danieldona.com/scratch>

1.8.10. EJEMPLO 9 (CICLOS):

Quiero que mi gato cuente de 1 a 10.

En este caso usaremos el bucle "repetir hasta que", dentro de la pestaña Control



Figura 51 EJEMPLO 9

Fuente: <https://www.danieldona.com/scratch>

Video:



El arte de programar es de verdad muy sistemático y organizado, podemos tener las ganas y las energías, pero sin una buena base y el correcto uso de las herramientas idóneas los sistemas que desarrollaremos no serán muy satisfactorias, en este proceso básico que estamos acudiendo será importante que el uso de las herramientas sea lo más intuitivo posible, SCRATCH nos entregara parte de esa intuición necesaria. Programación en SCRATCH

<https://www.youtube.com/watch?v=4ETghXmoUsw>

Profundización bibliográfica:



Se sugiere, antes de continuar, hacer lectura de los capítulos 1 del libro:

- » "GUIA DIDACTICA PARA PROFESORES SCRATCH".
<http://paginaspersonales.deusto.es/garaizar/scratch/doc/ScratchGu%C3%ADaDid%C3%A1cticaProfesores.pdf>
- » "MANUAL BASICO DE SCRATCH"
<http://www.ugelandahuaylas.gob.pe/portal/images/crt/LaptopXoSecundariaScratch.pdf?wmode=transparent>

IDEAS CLAVE:

El siguiente resumen podremos refrescar ideas importantes de SCRATCH

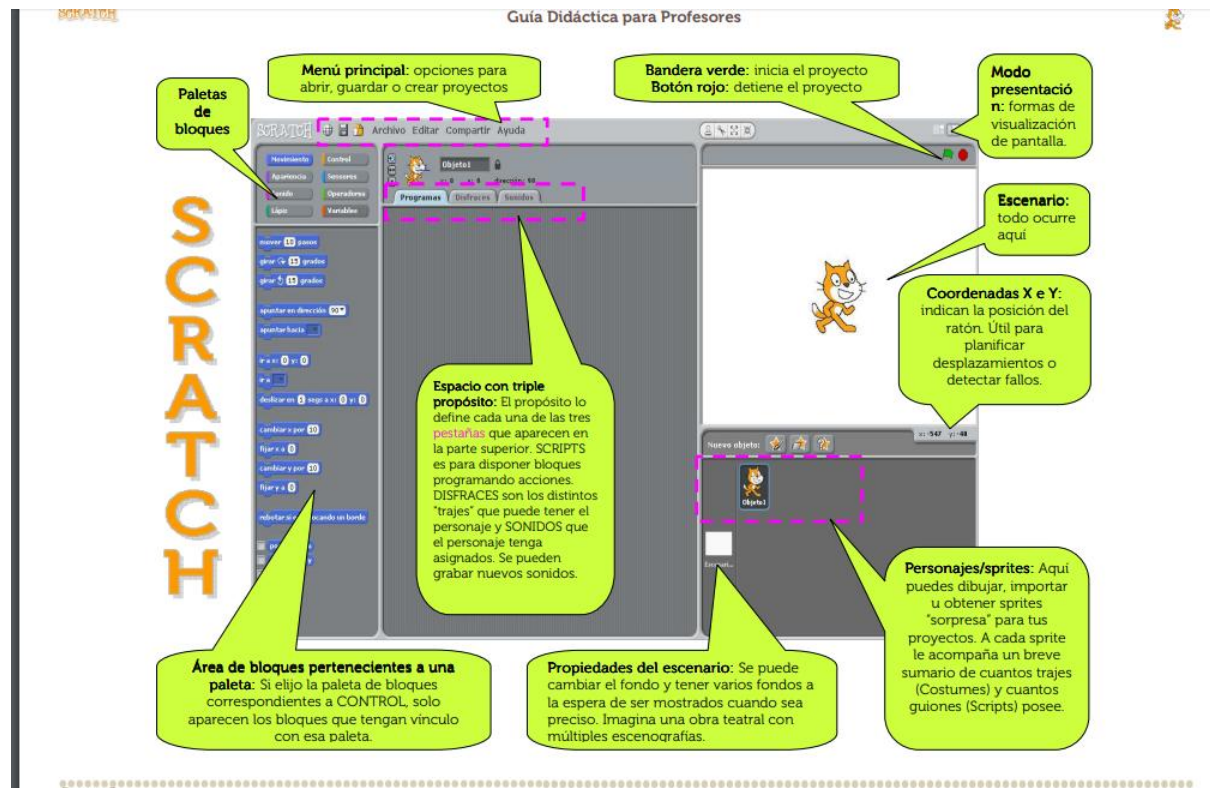


Figura 3 Entorno de Desarrollo de SCRATCH
FUENTE: www.paginaspersonales.deusto.es

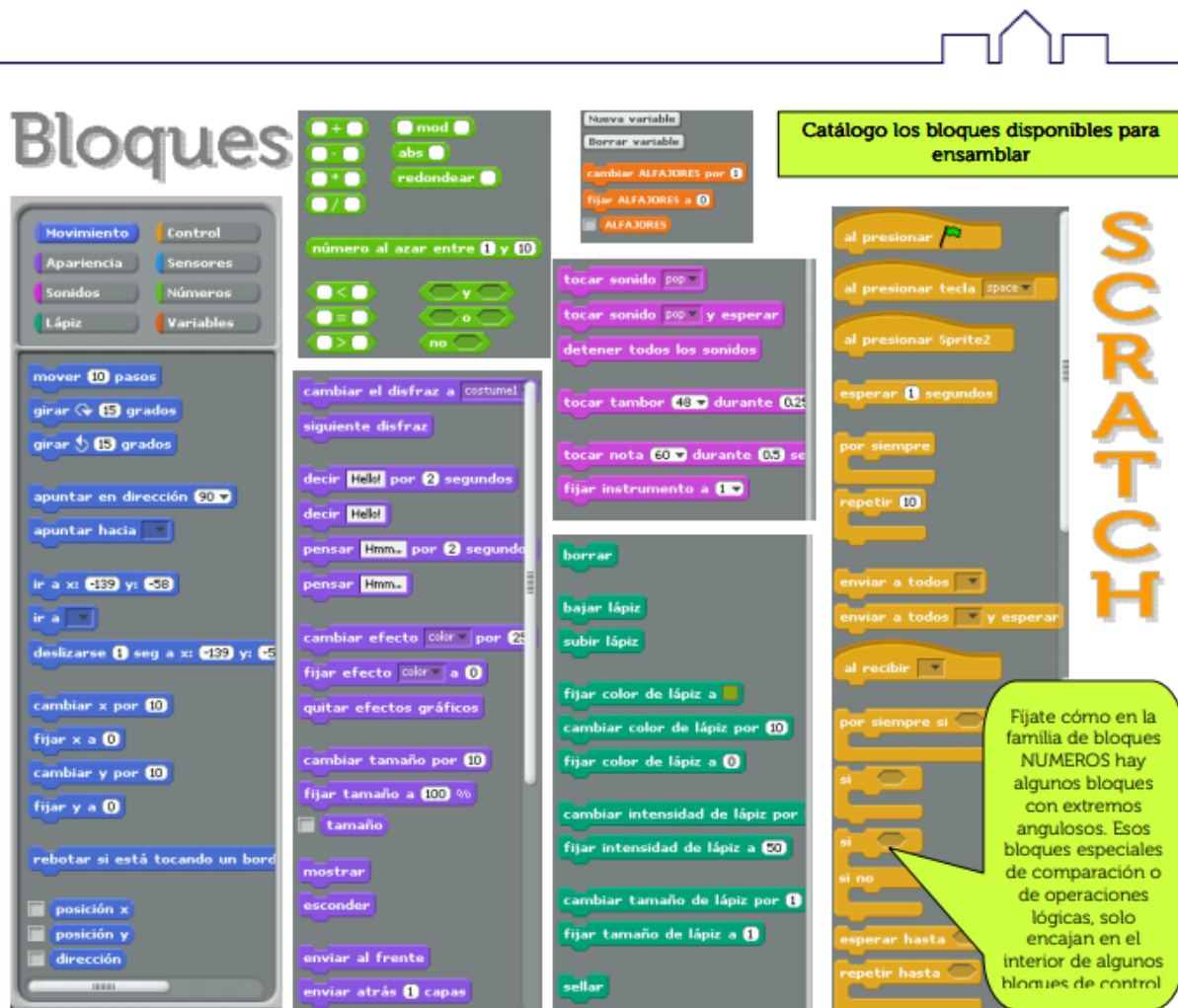




Figura 10 RESUMEN DE BLOQUES
FUENTE: www.paginaspersonales.deusto.es

LINKS/MATERIAL MULTIMEDIA

Espacio para indicar enlaces y/o materiales que complementen el tema visto. Es requisito fundamental que estos materiales sean de uso libre y debidamente referenciados. Explique qué va a encontrar el estudiante en ese recurso, para qué le sirve.

MÓDULO: "Fundamentos de Programación Computacional"		Unidad: 1
Recurso	Descripción	
 Video	En este video encuentras algunos de los elementos más importantes de la programación es SCRATCH Programación en SCRATCH https://www.youtube.com/watch?v=4ETghXmoUsw	
 Presentación	<ul style="list-style-type: none">• "GUIA DIDACTICA PARA PROFESORES SCRATCH". http://paginaspersonales.deusto.es/garaizar/scratch/doc/ScratchGu%C3%ADadaDid%C3%A1cticaProfesores.pdf• "MANUAL BASICO DE SCRATCH" http://www.ugelandahuaylas.gob.pe/portal/images/crt/LaptopXoSecundariaScratch.pdf?wmode=transparent	
 Lecturas complementarias	En el siguiente material podrás entender un poco más sobre Programación en SCRATCH http://www.ugelandahuaylas.gob.pe/portal/images/crt/LaptopXoSecundariaScratch.pdf?wmode=transparent	



CONCLUSIONES

En esta semana hemos podido comprender una herramienta de programación básica que hoy en día está siendo usada por muchas personas, las finalidades de esta herramienta son preferentemente aplicadas a un proceso básico de enseñanza que aportan un carácter práctico y además de un sentido gráfico que añaden elementos intuitivos al futuro desarrollador.

Pues bien, de ahora en adelante el sentido de estructuración: el paso a paso de la programación, sufrirá una leve desviación: La programación orientada a objetos nos entregará un paradigma de desarrollo en el cual la base del concepto se centrará en lo que es denominado: la clase y el objeto

Durante la próxima semana, veremos como la programación orientada a objetos se toma el mundo del desarrollo: Herramientas, conceptos y palabras claves son lo que aprenderemos en torno a este tipo de desarrollo.



BIBLIOGRAFÍA

- ✓ "GUIA DIDACTICA PARA PROFESORES SCRATCH".
<http://paginaspersonales.deusto.es/garaizar/scratch/doc/ScratchGu%C3%ADadid%C3%A1cticaProfesores.pdf>
- ✓ "MANUAL BASICO DE SCRATCH"
<http://www.ugelandahuaylas.gob.pe/portal/images/crt/LaptopXoSecundariaScratch.pdf?wmode=transparent>