

# GUÍA DIDÁCTICA CONTROL



## Pre-conocimiento

Hasta ahora ya hemos aprendido el entorno, a utilizar los bloques de movimiento, apariencia, sonido y lápiz.

En esta unidad vamos a ver los distintos tipos de bloques de control. Con su utilización podremos decidir en qué momento ejecutar un conjunto de bloques. Utilizar eventos como el clic del ratón o la pulsación de una determinada tecla. Así como estructuras repetitivas y comparaciones condicionales.



## Objetivos

- Conocer los más importantes bloques de cada una de las familias
- Realizar actividades para comprobar su uso
- Conseguir realizar nuestra práctica guiada de utilización de bloques de control
- Conocer las estructuras de control y repetición
- Saber qué es y para qué sirve una estructura condicional



## Temporalización

La propuesta temporal de esta unidad es estudiarla durante la semana:

Contenidos	Temporalización
Familia de bloques de control	1 sesión

# Contenidos



## Indice

1. Introducción
  2. Un poco de teoría
    1. El primer movimiento
    2. Descripción del funcionamiento de los bloques
    3. Los bloques uno a uno
    4. Eventos o sucesos
    5. Repeticiones o bucles
    6. Comparaciones
    7. Envío y recepción de mensajes
  3. Actividad guiada
  4. Fuentes de información
-

# 1. Introducción



## CONTROL



En esta sección encontraremos todos los bloques que permiten un control sobre la ejecución de las instrucciones. Así podremos repetir bloques, comparar números enviar mensajes a los objetos, etc.

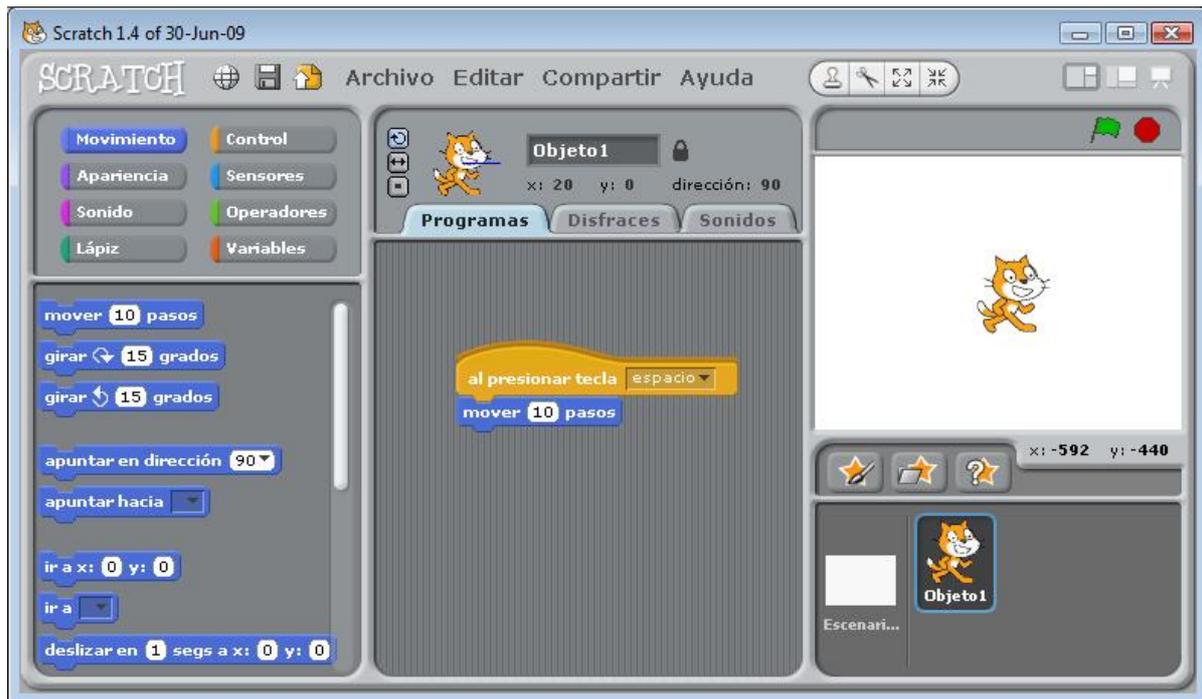
---

## 2. Un poco de teoría



### El primer control

A estas alturas del curso ya hemos utilizado en anteriores ejemplos y actividades algunas estructuras de control. Por ejemplo podemos cambiar el inicio de alguno de los programas que hemos hecho hasta ahora para que en lugar de ejecutarse al presionar la bandera verde lo hagan al presionar alguna tecla.



### Descripción del funcionamiento de los bloques

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

#### 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.



Si la condición es verdadera, ejecuta los bloques dentro de la porción **si**; si no, ejecuta los bloques que están dentro de la porción **si no**.



Espera hasta que la condición sea verdadera, para ejecutar los bloques siguientes.



Comprueba si la condición es falsa; si lo es, ejecuta los bloques en su interior y vuelve a chequear la condición. Si la condición es verdadera, pasa a los bloques siguientes.



Detiene el programa edl Objeto que contiene esta instrucción.



Detiene todos los programas de todos los Objetos.

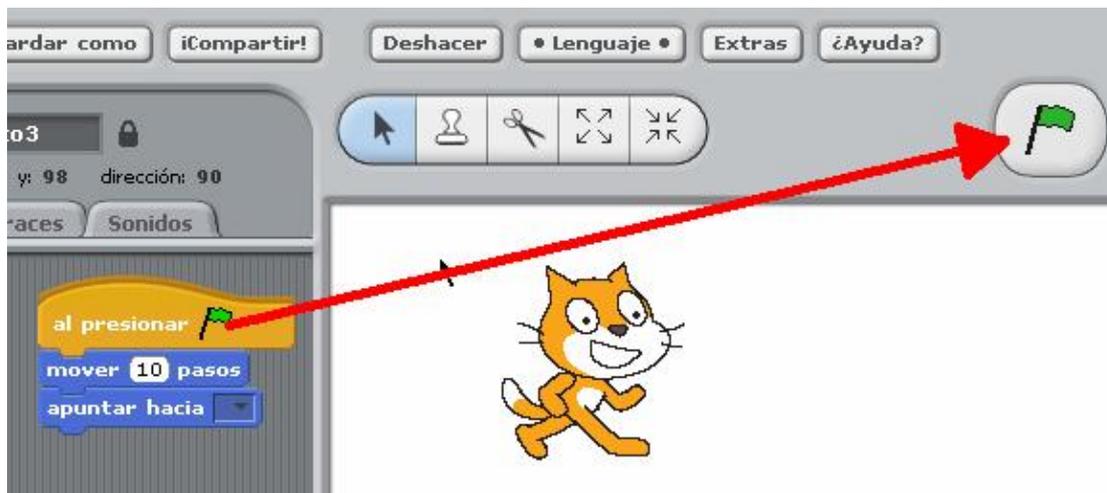


## LOS BLOQUES UNO A UNO

### El bloque al presionar (bandera verde)



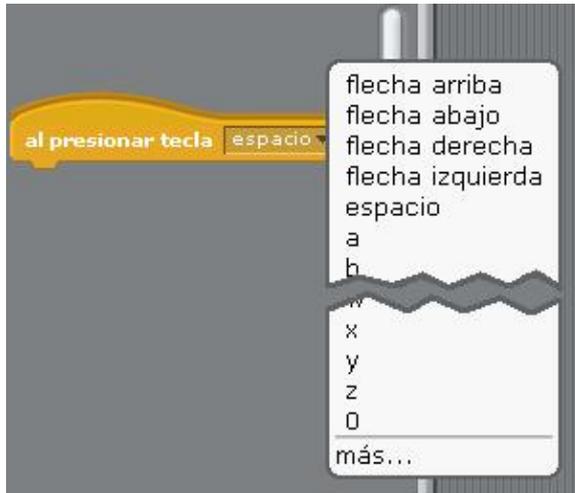
Este bloque desencadena automáticamente todas las instrucciones que tiene colgadas. Es importante saber que un mismo objeto puede tener varios scripts iniciándose con este bloque. Por lo mismo, varios objetos pueden recibir este bloque. Esto es absolutamente necesario si se quiere que varios objetos reaccionen al mismo tiempo cuando el programa se ponga en marcha.



### El bloque al presionar tecla \_\_\_



Este bloque funciona como el anterior, pero será una tecla la que desencadene la instrucciones en lugar de la bandera verde.



### El bloque al presionar Objeto\_



Las instrucciones se desencadenarán cuando el usuario clique sobre el objeto determinado.

Por ejemplo:



### Los bloques repetir por siempre y repetir \_\_ (número de veces)

Estos bloques hacen esencialmente la misma cosa: provocan un bucle infinito o de cierto número de veces de las instrucciones incluidas en ellos.



### El bloque esperar \_\_ segundos



Este bloque permite insertar una pausa de x segundos entre las instrucciones. Se puede asociar un número decimal como, por ejemplo, 0.003.

También es posible dar una fracción siempre que se utilice el bloque **división** de la categoría **NÚMEROS**.

### El bloque enviar a todos y enviar a todos y esperar



Este bloque permite la comunicación entre objetos. Así, un objeto puede enviar un mensaje a todos los demás que al recibirlo (ver el bloque al recibir) actuarán en consecuencia. Primero hay que clicar sobre el triángulo para que aparezca nuevo.



Al clicar sobre nuevo podremos escribir el título del mensaje. Aconsejamos escribir una sola palabra lo más relacionada posible con el contexto del mensaje.



El mensaje será escuchado por todos los objetos, pero no todos reaccionarán a él. Esto sólo se producirá con el bloque al recibir\_\_

Este bloque es idéntico al anterior con la diferencia que el objeto que lo envía esperará a que todos los receptores lo hayan recibido y hayan ejecutado sus instrucciones.

### El bloque al recibir\_\_



Este bloque permite a un objeto que recibe un mensaje ejecutar las instrucciones correspondientes. El pequeño triángulo permite elegir el mensaje. Podemos crear también un nuevo título. Evidentemente será preciso, en este caso, utilizar un objeto que envíe un mensaje con ese título.

### Los bloques condicionales



Los bloques condicionales combinan con la categoría de los **SENSORES** y la de los **NÚMEROS**, verificando que se produce una condición para que el script siga adelante. Un ejemplo de script: [Nótese que el bloque `¿tocando__?` está en la categoría **SENSORES**]



### Los bloques detener



El primer detiene un script en concreto. El segundo detiene todo tiene la misma función que el octógono rojo al lado de la bandera verde.



## EVENTOS O SUCESOS

Un evento o suceso es algo que "sucede" en un momento determinado. En esta caso puede ser la pulsación de una tecla, el pulsar sobre un personaje, etc. Scratch nos da la posibilidad de realizar acciones cuando se produzca un determinado evento. Gracias a esto, por ejemplo, podemos realizar cosas diferentes dependiendo de que tecla pulsemos.

Los bloques de control que se ocupan de esto son:



## REPETICIONES O BUCLES

Los bloques de repetición o bucle nos permite realizar un conjunto de bloques un número finito o infinito de veces. Esto evita que reescribamos una y otra vez lo mismo ahorrando trabajo y organizando mejor el programa final.

Los bloques encargado de esto son:



El bloque **repetir** \_\_ le podemos indicar un número determinado de veces o si nos fijamos en la forma de la pieza que podemos introducir, podríamos poner un bloque redondeado que corresponde como veremos más adelante a el valor de una variable, valor de un sensor, etc. en definitiva todo lo que tenga una forma redondeada en los extremos.



El bloque de **por siempre si** y **repetir hasta que** realizará la repetición mientras o hasta que se cumpla una cierta condición. El bloque para la condición tiene la forma de hexágono y lo denominaremos bloque lógico, estos bloques los veremos con más detenimiento en el apartado de operadores, pero vamos a ver como se utilizan brevemente.

Supongamos que tenemos definida una variable llamada **cont** .La crearemos de la siguiente forma:



Y nos aparecera esto:



Vamos ha mover nuestro personaje y cambiar de apariencia mientras nuestra variable tenga un valor inferior a 10. Para ello nos falta ver los bloques lógicos que tienen forma hexagonal. En concreto los de comparación son los siguientes:



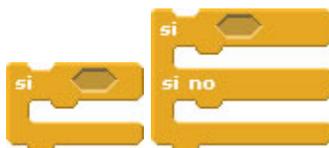
Nuestro pequeño programa quedaría así:



## COMPARACIONES

Los bloques de comparación nos permiten cambiar el orden de ejecución de los bloques dependiendo de si se cumple o no una condición ( bloque lógico).

Estos bloques son los siguientes:



La diferencia entre utilizar uno u otro está en si deseamos hacer una cosa cuando se cumpla y otra cuando se incumpla. El siguiente ejemplo tendría el mismo resultado porque dejaríamos por rellenar la parte del **si no**:

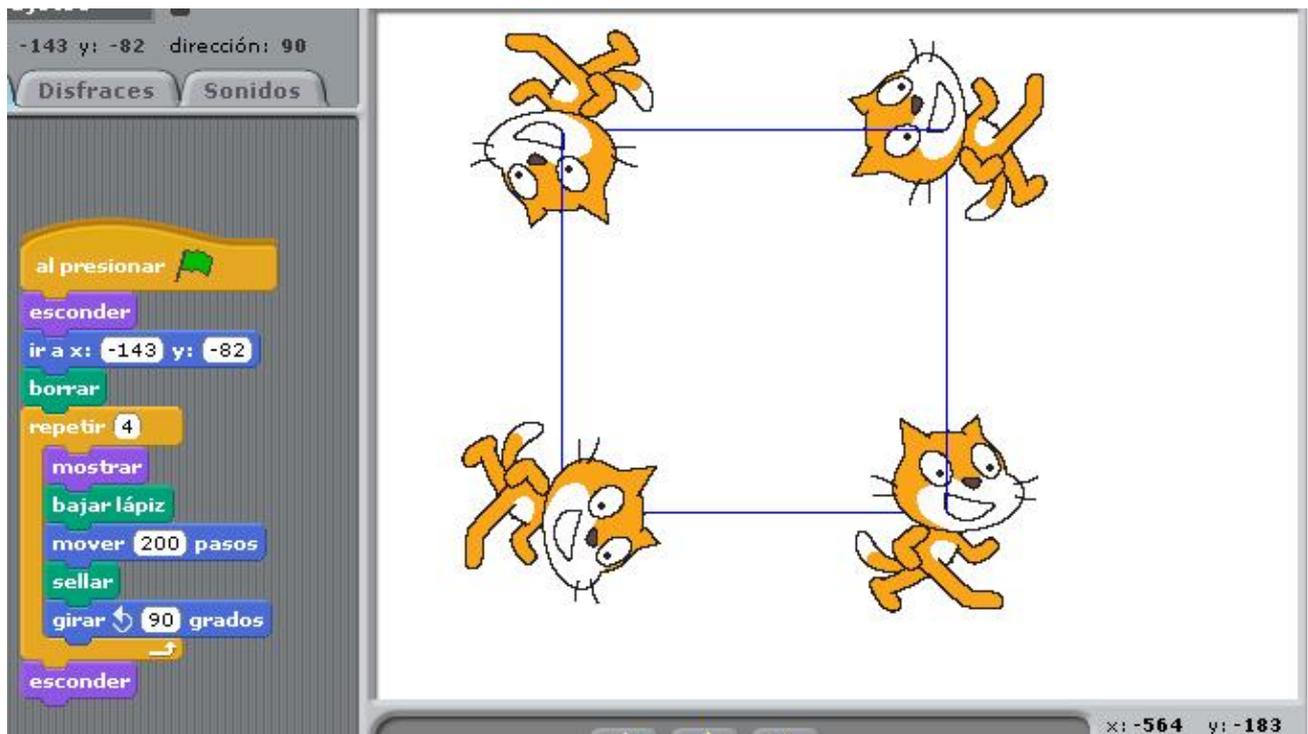


En caso de no cumplirse la condición todo los bloques de código que hallamos introducido en su interior no se ejecutarán. Puede haber una sola instrucción dentro o muchas.



## Un pequeño ejemplo

Pintar un cuadrado utilizando un proceso repetitivo y clonando en cada esquina el objeto.



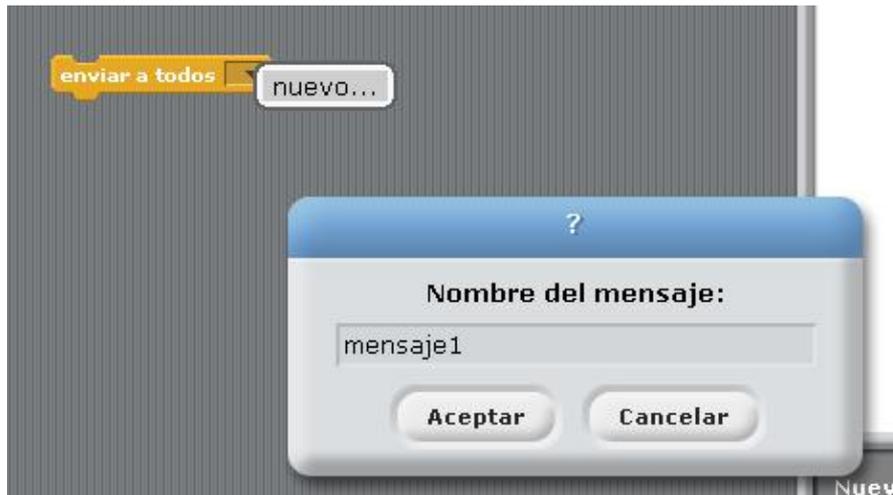
## ENVIO Y RECEPCIÓN DE MENSAJES

El uso de mensajes permite por una parte, la coordinación entre distintos objetos de

forma que podemos enviar en un momento dado desde un objeto un mensaje a otro para que realice alguna acción. Por otra parte también nos ayuda a mantener una estructuración en nuestro programa. Cuando nuestro programa crezca el poder separar conjuntos de bloques nos puede servir de ayuda. Estos mensajes no necesariamente han de enviarse de un objeto a otro distinto, pueden enviarse dentro del mismo objeto.

También es interesante recalcar que funciona como un evento, es decir que es independiente de la ejecución lineal del programa.

**Enviar a todos:** Al seleccionar la pestaña seleccionamos la variable o mensaje existente o generamos uno nuevo. Se nos abre una cajeta y tecleamos el nombre del mensaje a enviar o variable (argumento) global.



La diferencia entre **enviar a todos** y **enviar a todos y esperar** es que, en la primera, enviamos el mensaje y seguimos ejecutando el bloque de código que sigue; mientras que, en el segundo, enviamos el mensaje y, mientras no se ejecute en algún objeto que lo reciba, no se seguirá ejecutando el código desde donde lo enviamos.



Ya sólo nos queda escribir los bloques que queremos que se ejecuten cuando se reciba el mensaje. Para ello cambiaremos por ejemplo el iniciar el programa con la bandarita verde por iniciar el programa al recibir un mensaje determinado.



### Ejemplo de uso de este bloque:

Al presionar un objeto2 enviar un mensaje y ejecutar el código en el objeto 2.

**Objeto1**

**Objeto2**



Al presionar la banderita verde se envía el mensaje desde el objeto 1 mensaje1 enviado desde el objeto1, el objeto2 lo recibirá y lo que tarda en ejecutarse el código mostrará en pantalla dos en el objeto2 (derecha) para seguir ejecutando los bloques girar, decir y fijar tamaño.

## 3. Actividad guiada



### Actividad guiada

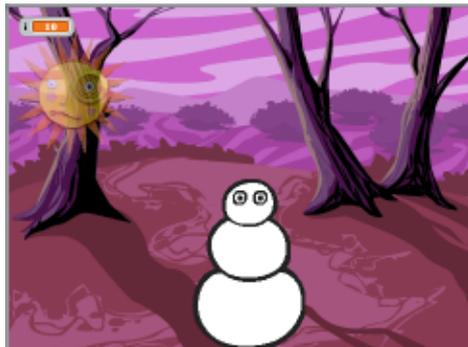
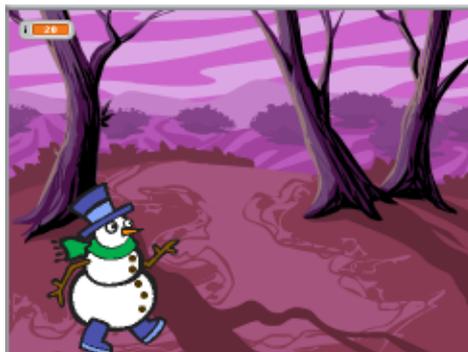
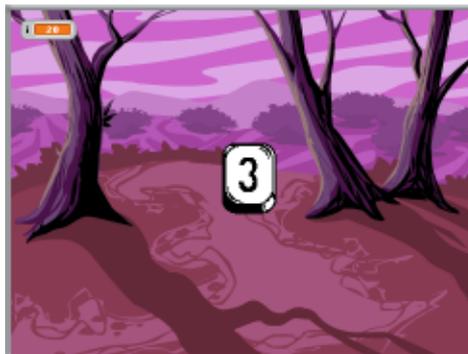
## Control. (Bloque Naranja)

La idea de la presente práctica es utilizar las diversas opciones que nos brinda el bloque control en Scratch.

Para realización de esta práctica vamos utilizar:

- 1 escenario con 2 fondos
- 3 objetos (uno con un disfraz, el segundo con 2 disfraces y el tercero con 3). Tanto los objetos como el fondo tienen código de programa asociado.

En la actividad guiada vamos a generar una pequeña animación guiada por paso de mensajes entre objetos. Este es un pequeño resumen de lo que verás después en tu programa cuando termines la actividad:





La sucesión de mensajes que se envían entre objetos para llevar a cabo la animación son:



Sigue los siguientes pasos para completar la animación. Es interesante que compruebes cómo se va desarrollando la secuencia del programa mediante la utilización del paso de mensajes.

**PASO 1: Objeto escenario. Fondos: woods y desert.**

Modifica el escenario para que contenga los siguientes fondos:



Introduce el siguiente código asociado al escenario:

### Bloques de código

### Detalle del código generado



- Al iniciar el proyecto cambiamos al fondo "woods" y enviamos el evento "mostrar3".



- Al recibir el evento "desierto" cambiamos al fondo "desert" mediante el efecto de desvanecer.

## PASO 2: Objeto nieve. Disfraces: snowman1 y snowman2

Crema el objeto nieve y añade los disfraces snowman1 y snowman2 de la siguiente forma:



Copia el siguiente código asociado al objeto nieve:

### Bloques de código



### Detalle del código generado

- Al inicio escondemos el objeto.
- Al recibir el mensaje "nieve1" desplazamos el objeto a la parte central del escenario mediante una instrucción "repetir". Cambiamos el disfraz y enviamos el mensaje "sol".
- Al recibir el mensaje "marchar" cambiamos el disfraz del objeto, lo desplazamos hacia la parte derecha del escenarios mediante ua instrucción "repetir". Por último, detenemos la ejecución de todo el programa.

### PASO 3: Objeto número. Disfraces: números 1, 2 y 3

Crea el objeto número con los siguientes disfraces:



No es necesario que sean estas imágenes de números, puedes utilizar otras formas que aparezcan en Scratch o crearlos tu mismo, pero si que debes de respetar el orden y los nombres.

Ahora copia el siguiente código asociado al objeto:

#### Bloques de código



#### Detalle del código generado

- Al inicio escondemos el objeto.
- Al recibir el mensaje "mostrar3" cambiamos al disfraz del número 3 y enviamos el mensaje "mostrar2".



- Al recibir el mensaje "mostrar2" cambiamos al disfraz del número 2 y enviamos el mensaje "mostrar1".

- Al recibir el mensaje "mostrar1" cambiamos al disfraz del número 1 y enviamos el mensaje "nieve1".

#### PASO 4: Objeto sol. Disfraz: sun

Por último crea el objeto sun con el siguiente disfraz:



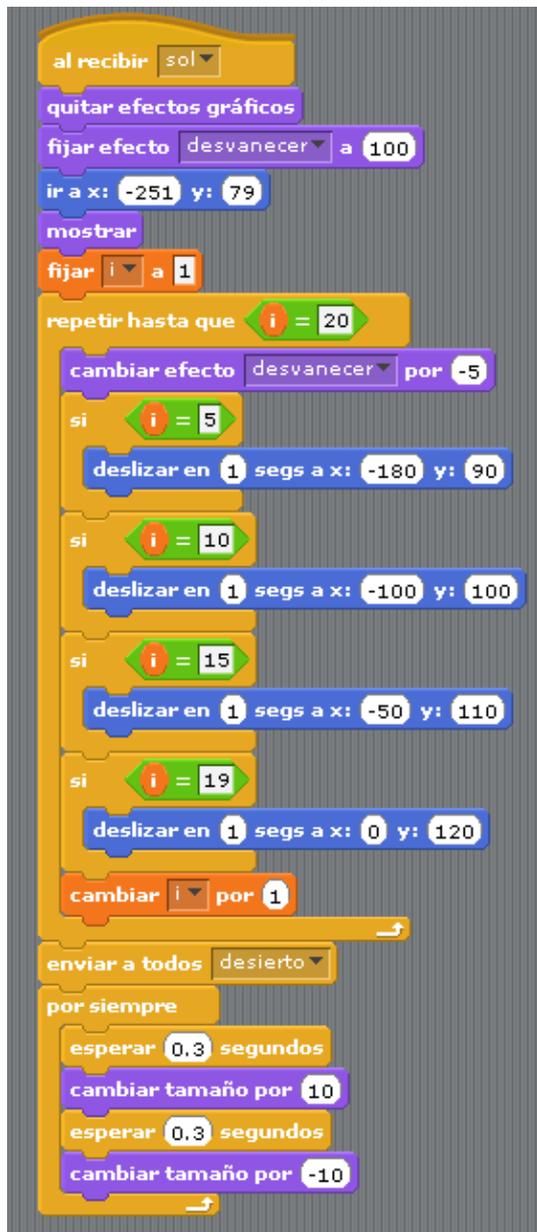
Ahora copia el siguiente código:

#### Bloques de código



#### Detalle del código generado

- Al inicio escondemos el objeto.



- Al recibir el mensaje "sun" hacemos aparecer el objeto por la parte izquierda del escenario.
- Hacemos deslizar y aparecer poco a poco el sol mediante el efecto "desvanecer". Se utiliza para ello la instrucción "repetir hasta que". **Para este bloque de código hay que crear una variable i.**
- Enviamos el mensaje "desierto".
- Por último con una instrucción "por siempre" hacemos que el sol simule que está palpitando.

Comprueba el funcionamiento de la animación y viendo el código **intenta entender cómo se ha utilizado la sucesión de mensajes** indicados al principio para dirigir toda la animación.

Por último, guarda y envía la actividad guiada con el nombre de **ActGuiadaCon**.

## 4. Fuentes de información



### Fuentes de información

- “Reference Guide – SCRATCH, version 1.3”, elaborado por el grupo de investigación del <http://www.media.mit.edu/> MIT Media Lab y traducido al español al español por Eduteka.
  - Guia de referencia de CEIP de Cella (Teruel), 2009.
  - <http://aulavirtual2.educa.madrid.org/course/view.php?id=665>
-