

## De espectador a programador: El camino para entender la tecnología aprendiendo a manejarla



### Bloque 3. Comunicación entre objetos y variables

#### Autores

Eva María Castro Barbero

Jesús M. González Barahona

Gregorio Robles Martínez

Pedro de las Heras Quirós

## Índice

Comunicación entre objetos y variables.....	2
Objetivos .....	2
Descripción de contenidos y vídeos .....	2
Coordinación entre objetos mediante mensajes.....	3
Envío y recepción de mensajes.....	4
Espera condicional .....	5
Variables.....	6
Preguntas: interacción con el usuario .....	9

## Comunicación entre objetos y variables

### Objetivos

Cada objeto tiene sus propios programas. Todos los objetos ejecutan su código a la vez. Pero en ocasiones queremos que los objetos interactúen, para lo cual necesitamos aprender mecanismos que nos permitan especificar los patrones de comunicación entre los objetos y que permitan sincronizar sus acciones.

A veces necesitamos almacenar información que los objetos necesitarán utilizar más tarde.

Los objetivos principales de este bloque son:

1. aprender a sincronizar las acciones de los programas de diferentes objetos de forma que podamos programar comportamientos coordinados de los objetos.
2. aprender a utilizar las variables como mecanismo de almacenamiento de información que puede ser recuperada más tarde.

### Descripción de contenidos y vídeos

En este bloque aprenderemos a coordinar las acciones de diferentes programas mediante el envío de mensajes. Posteriormente aprenderemos a crear y utilizar variables.

Para una mejor comprensión de este bloque, carga la página de Scratch (<http://scratch.mit.edu>) en otra ventana/pestaña de tu navegador y entra con tu nombre de usuario y contraseña. A la vez que vas leyendo esta guía puedes deberás ir practicando, realizando dentro de tu página de Scratch las mismas acciones que se describen en esta guía. Ten en cuenta que una de las actividades de este bloque consiste en continuar el proyecto que se realiza con esta guía paso a paso.

Para estudiar este bloque de contenidos puedes optar por visualizar un conjunto de vídeos o por leer la documentación escrita que encontrarás a continuación. Los contenidos de los vídeos y la documentación escrita son los mismos. Te recomendamos que veas los vídeos y si tienes alguna duda concreta repases la documentación escrita:

- [\*\*Vídeo 1: Motivación de la necesidad de coordinar las acciones entre objetos \(6:37 minutos\)\*\*](#)

El vídeo utiliza un ejemplo para motivar la necesidad de utilizar mecanismos de coordinación de objetos para que sus acciones se puedan sincronizar de manera segura. Usar instrucciones para esperar un tiempo no es la mejor solución para coordinar acciones de objetos.

- [\*\*Vídeo 2: Coordinación de objetos mediante envío y recepción de mensajes \(9:40 minutos\)\*\*](#)

El vídeo muestra como el anterior ejemplo se puede programar de manera más fiable utilizando el envío y recepción de mensajes, un mecanismo que sirve para sincronizar las acciones de diferentes objetos, que es explicado mediante un ejemplo en este vídeo.

- [\*\*Vídeo 3: Coordinación de objetos mediante "esperar hasta que" \(4:20 minutos\)\*\*](#)

El vídeo muestra cómo la instrucción "esperar hasta que" puede en ocasiones servirnos también para coordinar las acciones de varios objetos. Se utiliza un ejemplo en el que también se usa la instrucción ya vista "repetir hasta que".

- [Vídeo 4: Variables \(14:39 minutos\)](#)

El vídeo introduce el concepto de la "variable" mediante un ejemplo sencillo. Las variables son unos mecanismos que podemos utilizar para guardar valores (números, textos) y recordarlos posteriormente en el programa. El vídeo utiliza la analogía de una nota pegada en la nevera para recordar la lista de la compra.

- [Vídeo 5: Interacción con el usuario \(03:46 minutos\)](#)

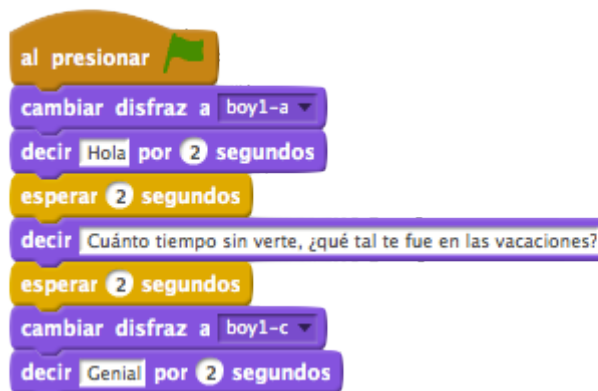
## Coordinación entre objetos mediante mensajes

Los objetos y fondos de un proyecto pueden comunicarse entre sí y coordinarse mediante el envío y recepción de mensajes: un objeto/fondo envía un mensaje que puede ser recibido por cualquier otro objeto y fondo.

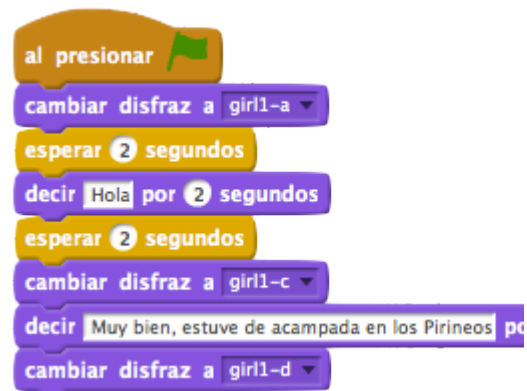
Uno o más objetos/fondos pueden reaccionar a la recepción de un mensaje, ejecutando un programa.

Por ejemplo, si deseamos que dos objetos mantengan un diálogo, podríamos utilizar las instrucciones "decir por segundos" y "esperar segundos" para coordinar el diálogo entre ambos. Pulsa sobre [este proyecto](#) para ver cómo está programado.

**Programa del Objeto Chico**



**Programa del Objeto Chica**



La coordinación entre ambos objetos se realiza controlando el tiempo que tiene que esperar cada objeto para mostrar el siguiente diálogo.

Observa que mientras el objeto Chico está mostrando un diálogo, el objeto Chica debe estar esperando.

Para que el control del diálogo sea más fácil y no haya que estar calculando el número de segundos que cada uno tiene que esperar, un objeto puede avisar a otro de que ha terminado de mostrar su diálogo a través del envío del evento mensaje. El otro objeto estará esperando recibir este evento y cuando lo reciba sabrá que el otro objeto ha terminado de mostrar su cuadro de diálogo y ahora le tocará a este objeto mostrar su diálogo.

Pulsa sobre [este proyecto](#) que también desarrolla el mismo diálogo entre los dos personajes pero utilizando el evento de envío y recepción de mensajes.

En este nuevo proyecto cada vez que un objeto termina de mostrar su diálogo envía un mensaje al otro objeto.

### Programa del Objeto Chico



### Programa del Objeto Chica



## Envío y recepción de mensajes

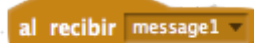
El envío y recepción de mensajes son dos instrucciones de control del tipo **Eventos**.

La instrucción "enviar message1" provoca que se envíe el mensaje denominado "message1" a todos los objetos. El nombre del mensaje hay que definirlo en la instrucción, al crear un nuevo mensaje.



Cuando tenemos un programa que envía varios mensajes, es recomendable definir nombres que ayuden a saber a qué mensaje se refiere en cada momento.

La instrucción "al recibir message1" sólo puede aparecer como la primera instrucción de un programa. Fíjate en la forma de esta instrucción y en cómo no se pueden enganchar bloques encima de éste:



Cuando el mensaje especificado es recibido, en este ejemplo "message1", las instrucciones que aparecen a continuación se ejecutan.

Pinchando con el botón derecho del ratón sobre el nombre del mensaje se puede saber qué objetos envían el mensaje y cuáles lo reciben, los objetos aparecen redondeados en amarillo en la zona de objetos.

En algunos casos puede ser útil enviar un mensaje y esperar a que todos los objetos lo hayan recibido. El siguiente bloque "enviar y esperar" permite enviar un mensaje a todos los objetos y fondos del proyecto y además espera hasta que todos los elementos han recibido dicho mensaje:



El uso de mensajes nos permite coordinar las ejecuciones de los programas de diferentes objetos, asegurándonos de que un programa comience a ejecutarse sólo después de que otro programa haya ejecutado determinadas instrucciones.

## Espera condicional

La espera condicional es una instrucción que detiene la ejecución de un programa hasta que una determinada condición sea verdadera. Cuando dicha condición sea verdadera el programa continuará la ejecución con la siguiente instrucción. Este tipo de instrucción también se utiliza para coordinar acciones entre objetos.

La espera condicional se programa con la instrucción "esperar hasta que " de las instrucciones de tipo **Control**.



Por ejemplo podemos usar esta instrucción para esperar que un objeto esté tocando a otro y en ese momento realizar otra acción. Pulsa sobre el [siguiente proyecto](#) para ver un ejemplo. En este caso cuando el tiburón alcanza al pez, el pez se escapa. El programa del pez está esperando a que el tiburón le toque (objeto Shark), utilizando para ello la instrucción "esperar hasta que ", cuando esta condición es verdadera, el pez se mueve a otra posición y escapa.

Éste es el programa del pez:



## Variables

Las variables se utilizan para recordar valores.

Vamos a pensar en cómo haríamos el siguiente programa en Scratch: queremos contar en número de veces que un usuario pulsa con el ratón sobre un objeto del escenario. Cada vez que el usuario pulse sobre el objeto, el objeto mostrará un diálogo que indique el número de veces que ha sido pulsado.

Para ello, crea un nuevo proyecto que por defecto tendrá el objeto Gato y utiliza el bloque "al clicar este objeto" de la siguiente forma:

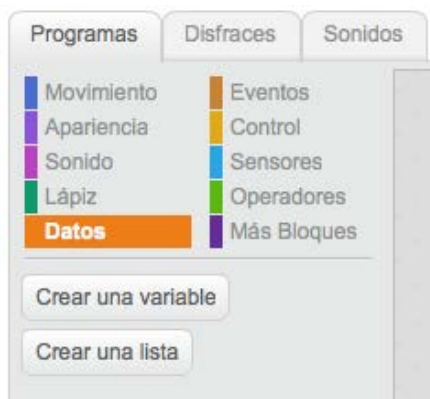


Con este programa, cada vez que se pulse sobre el Gato, aparecerá el mensaje "Me has pulsado 1". Sin embargo, si deseamos que muestre el número de veces que se ha pulsado durante la ejecución del proyecto, deberíamos mostrar la primera vez: "Me has pulsado 1", la segunda vez: "Me has pulsado 2", la tercera vez: "Me has pulsado 3", y así sucesivamente.

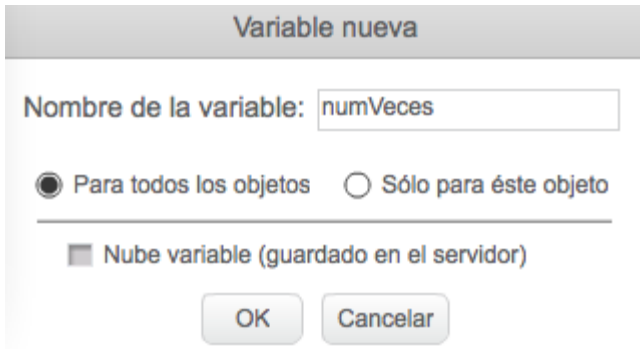
Para realizar este programa necesitamos recordar el número de veces que ha sido pulsado el Gato, de tal forma que cada vez que pulsemos sobre dicho objeto sumemos uno a ese número de veces y mostremos un mensaje en el que aparezca dicho número de veces.

Para recordar el número de veces que hemos pulsado un objeto vamos a crear una variable, una variable permite guardar un valor de algo, ya sea un número o una cadena de texto. En nuestro caso vamos a crear una variable para guardar el número de veces que ha sido pulsado el objeto. Vamos a llamar a esa variable numVeces.

Desde el menú de Programas, seleccionamos las instrucciones de tipo **Datos**:



Pulsamos sobre el botón "Crear una variable". En la casilla "Nombre de variable" le vamos a asignar un nombre, en nuestro caso "numVeces", dejaremos marcada la casilla que esté marcada por defecto ("Para todos los objetos") y pulsaremos sobre el botón OK.



Observarás como han aparecido en las instrucciones de **Datos** los siguientes bloques asociados a la variable nueva que acabamos de crear.



También verás como en el escenario aparece un monitor con el contenido de la variable "numVeces", su valor inicial es 0.





El monitor se puede ocultar o mostrar. Mientras estemos realizando el programa es útil mostrar el monitor para ver el contenido de nuestra variable. Para mostrar/ocultar el monitor de la variable "numVeces" se pulsa sobre la siguiente instrucción que se muestra en **Datos**:



Si está seleccionado, se mostrará el monitor de la variable "numVeces". De momento vamos a dejarlo seleccionado para ver cómo va cambiando.

El resto de instrucciones que han aparecido en el menú **Datos** se utilizan de la siguiente forma:



Asignar un valor a la variable numVeces. En este caso se está asignando el valor 0.



Si el contenido de la variable es numérico, sumo un determinado número al contenido de la variable. Este número puede ser positivo o negativo. En este caso suma 1 al valor que tenga asignado numVeces.



Muestra el monitor de la variable numVeces.

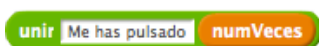


Oculto el monitor de la variable numVeces.

Vamos a usar el bloque "cambiar numVeces por 1" para que al pulsar sobre el objeto, la variable numVeces aumente en una unidad.



Cada vez que pulsamos sobre el objeto, podremos ver como en la variable numVeces que se muestra en el monitor, va aumentando. Sin embargo, el diálogo que muestra el objeto siempre dice que le hemos pulsado una vez. Nos gustaría que mostrara en el diálogo el número de veces total que hemos pulsado el objeto. Para ello vamos a unir el texto "Me has pulsado" con el número de veces que hemos pulsado, es decir, el contenido de la variable numVeces. Utilizaremos el bloque "unir" del tipo de instrucciones **Operadores** para concatenar el texto "Me has pulsado" y el contenido de la variable numVeces. Para acceder al contenido de la variable numVeces, arrastraremos el bloque con el nombre de la variable que aparece en las instrucciones de tipo **Datos** sobre el bloque "unir":



Y ahora arrastraremos este bloque que concatena esos dos valores sobre el bloque "decir por segundos":

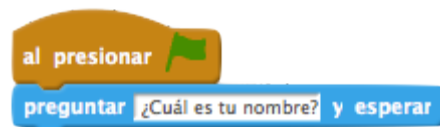


Al ejecutar nuevamente este programa, observarás ahora como el diálogo muestra cómo cambia el número de veces que se pulsa sobre el objeto. Sin embargo, la cuenta del número de veces no se ha iniciado en 1, si no que ha continuado con el valor que tenía de la ejecución anterior. Para que comience en 1 cada vez que se ejecuta el programa es necesario añadir la inicialización de esta variable, es decir, asignarle un valor inicial.



## Preguntas: interacción con el usuario

El bloque "preguntar y esperar" del conjunto de instrucciones **Sensores**, permite que los objetos puedan establecer un diálogo con la persona que está ejecutando el proyecto. Por ejemplo, podemos usar este bloque en el siguiente programa:



Al ejecutar este programa, en el escenario aparecerá en la parte inferior del escenario un cuadro de texto donde podremos introducir la respuesta:



El texto que introduzcamos quedará almacenado en una variable del sistema, llamada respuesta, a la que se puede acceder a través de las instrucciones **Sensores**. Podemos utilizar el contenido de esta variable para ofrecer una respuesta personalizada al usuario, por ejemplo:



Este programa solicita el nombre al usuario y construye un saludo personalizado con el nombre del usuario.



Formación en Red del INTEF

Licencia [Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).