

---

# Table of Contents

Introducción	1.1
--------------	-----

---

## 1. Toma de contacto

Pensamiento computacional	2.1
¿Qué herramientas tenemos?	2.2
Instalación	2.3
On-Line	2.3.1
Off-Line	2.3.2
Entorno Scratch	2.4
¡Ya funciona!	2.5

---

## 2. Empezando a manejar Scratch

Empezando a manejar Scratch	3.1
Bloques	3.2
Objetos, Disfraces y Fondos	3.3
Programas y Movimiento	3.4
Lápiz y Sonido	3.5
Control: condiciones	3.6
Control: repeticiones	3.7
Mensajes	3.8
Variables	3.9
¡El primer juego!	3.10

---

## 3. Proyecto

Módulo 3 Proyecto	4.1
Movimiento pez	4.2
Movimiento tiburón	4.3
Appearance	4.4
Sonido	4.5
Me largo!!!	4.6
Cobarde!!!	4.7
Dos jugadores	4.8
Consejos	4.9
Créditos	4.10

---



## Introducción

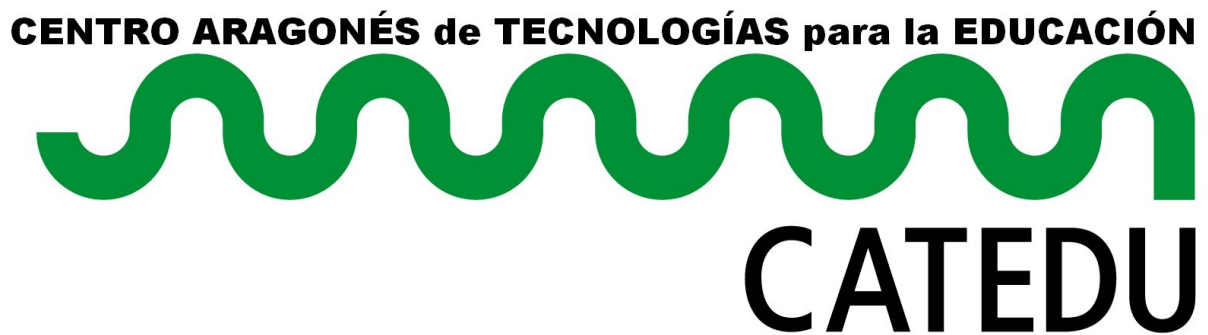


Scratch nació en el ámbito informal para compartir proyectos, música... y en el 2003 con el Instituto de Tecnología de Massachusetts (MIT) y Playful Invention Company se creó la primera versión de escritorio con el propósito de ayudar a niños **a partir de 8 años** a aprender a programar. En el 2013 salió la versión web.

En definitiva Scratch es una herramienta que permite de una forma rápida, sencilla e intuitiva, llevar el **pensamiento computacional** a la educación con niños.

## Objetivos

- Conocer los beneficios del pensamiento computacional en la educación
- Conocer las instrucciones básicas de programación
- Gamificación: Aprender jugando con proyectos



# Pensamiento computacional

El origen del término Pensamiento Computacional fue marcado por Jeannette M. Wing. El pensamiento computacional está directamente relacionado con la informática. Representa una aptitud aplicable universalmente para resolver problemas, diseñar de sistemas, pensar de forma paralela y recursiva. Implica conceptualización, abstracción y descomposición de tareas complejas. Jeannette M. Wing, debido a todas las ventajas que aporta el pensamiento computacional, propone la inclusión de este tipo o de formación en edades tempranas.

J. M. Wing. 2006. Computational Thinking. Communications Of The ACM. 49 (3).

<https://www.cs.cmu.edu/~CompThink/papers/Wing06.pdf>

**TENGAMOS EN CUENTA ANTES DE EMPEZAR...**

**...QUE EL PENSAMIENTO COMPUTACIONAL PUEDE LLEGAR A SER MUY VARIADO ENTRE DOS PERSONAS.**

En este curso vamos a ir construyendo la programación **de un juego paso a paso**. Iremos incluyendo bloques partiendo de las necesidades más simples y directas, luego las iremos completando según necesidades más ambiciosas. La secuencia de creación del juego y de programación de los bloques en este curso es bastante lógica y racional.

Pero hay que tener en cuenta que el orden de pensamiento computacional para crear programación puede realizarse de formas muy diferentes. En la realidad, es difícil **que dos personas, de forma independiente y autónoma, sigan exactamente los mismos pasos** para crear un mismo programa siguiendo una misma lógica y secuencialización, ¡Incluso llegando a la misma solución, el orden de razonamiento de ambas personas puede ser muy diferente!

La lógica de razonamiento de cada persona es diferente y esto hace que no haya una forma correcta de programar, sino tantas formas correctas como personas hayan dado con una de las posibles soluciones al problema. También es cierto que habrá soluciones que sean más simples que otras, o que funcionen mejor que otras. Y esto también es un aspecto a valorar y a tener en cuenta en cuanto a una posible evaluación de una estructura de programación.

Es posible que algún estudiante de este curso abordara la solución a este juego de formas muy diferentes, por ejemplo, empezando por los pasos finales de este curso y acabando por los pasos iniciales. Por tanto, si os veis con capacidades para ir construyendo el juego, podéis intentar construir vosotros mismos las soluciones. Si por contra os resulta difícil imaginar qué hacer en cada paso, os podéis dejar guiar por el razonamiento y secuencialización mostrado en este curso.

Es de esperar que al intentar realizar el siguiente proyecto de programación, cada vez seáis más autónomos. No hay que olvidar que para aprender a aplicar el pensamiento computacional para construir aplicaciones o juegos, hay que enfrentarse de forma individual ante nuevos retos, probar diversas alternativas y acabar dando con la solución de forma autónoma, lo cual incrementará la satisfacción y entrarán ganas de afrontar un reto mayor de programación.

**Y DE CARA A IR CONSTRUYENDO EL JUEGO...CADA VEZ QUE AÑADAS UNOS BLOQUES, PON EN MARCHA EL JUEGO PARA PROBAR QUE HACEN LO QUE ERA DE ESPERAR.**

## Pregunta Verdadero-Falso

Responde a la siguiente pregunta

### Pregunta 1

El pensamiento computacional se asocia y se relaciona únicamente con la informática.

☐ Verdadero ☐ Falso

Solución

**Respuesta correcta: Falso**

La influencia del pensamiento computacional también tiene influencia en otras disciplinas científicas: robótica, estadística, biología, física, química, economía, e incluso existen ejemplos de uso de Scratch en proyectos de letras y artes.

# RoboTICa

Oferta de formación en Pensamiento computacional del Centro Aragonés de Tecnologías para la Educación.



## ¿Qué herramientas tenemos?

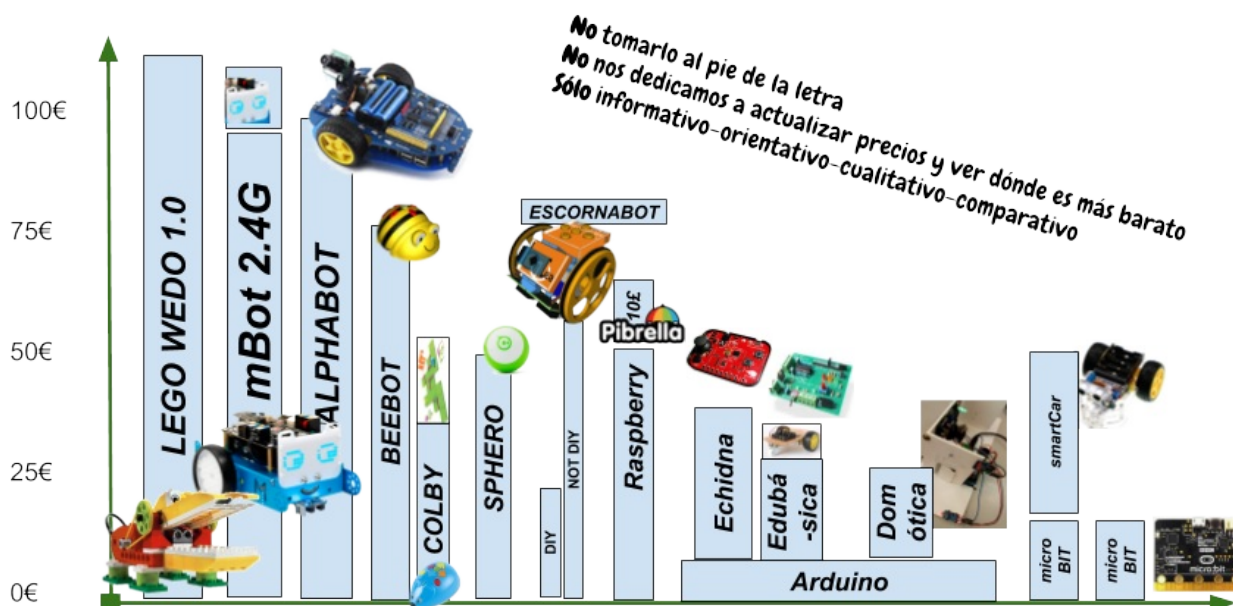
- App Inventor: <http://appinventor.mit.edu/explore/>
- Code: <https://code.org/>
- Blockly: <https://developers.google.com/blockly/>
- <https://blockly-demo.appspot.com/static/demos/index.html>
- <https://blockly-games.appspot.com/?lang=es>

## Propuesta de pensamiento computacional

Hay muchas herramientas, pero no todas son válidas ni a cualquier edad. En CATEDU hemos elaborado esta hoja de ruta seleccionando algunas y clasificándolo por edades (no se tiene que tomar al pie de la letra, sólo a nivel orientativo) para trabajar el pensamiento computacional, podríamos elegir otros modelos de robots, pero estos son los que trabajamos en CATEDU. **Puedes ver dónde encaja Scratch:**



### Coste aproximado de los packs de robótica



"No esperamos que todos los niños que aprenden a escribir se conviertan en novelistas, ni que todos los que estudian álgebra acaben siendo matemáticos; sin embargo, ambas son consideradas como habilidades fundamentales que todos los niños deben aprender. La programación informática también debería serlo",

[Hadi Partovi](#) cofundador de [Code.org](https://code.org), una organización sin ánimo de lucro que promueve la enseñanza de código en las escuelas en EE UU.

Como ves SCRATCH JUEGA UN PAPEL MUY IMPORTANTE en toda la primaria y secundaria, y **CODE.ORG** es una herramienta muy buena como base si no se quiere empezar de lleno en Scratch

También vemos que SCRATCH es un lenguaje apropiado para la robótica



# Instalación

## Dos formas de trabajar en SCRATCH

La web oficial de Scratch es <http://scratch.mit.edu/>

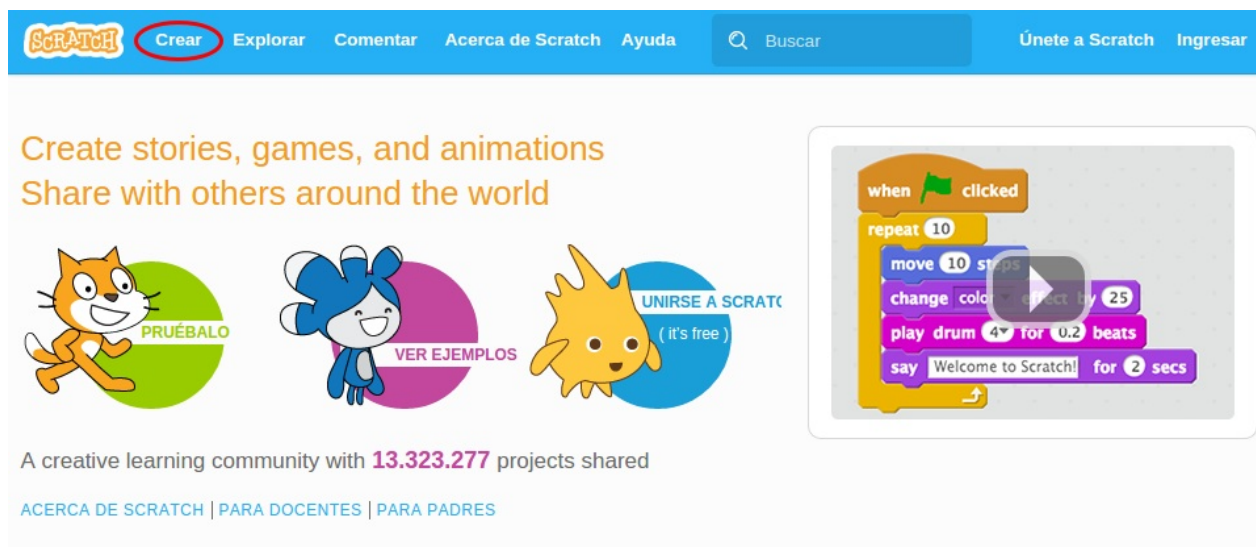
Hay dos caminos para crear proyectos con Scratch 2:

### 1. Sin necesidad de conexión a Internet: mediante Scratch 2 Offline Editor

- Es el editor de proyectos Scratch que no necesita conexión a Internet para funcionar.
- Recomendada en caso de existir problemas de conexión.
- Los proyectos Scratch se guardan en archivos con extensión .sb2 dentro de nuestro ordenador.
- [Web oficial de descarga de Scratch 2](#)

### 2. Con necesidad de conexión a Internet: mediante Scratch 2 Online

- Se realiza desde la página oficial <http://scratch.mit.edu/> en CREAR. (Acceso directo: <https://scratch.mit.edu/projects/editor/>)
- Los proyectos Scratch se guardan en la Web de Scratch <http://scratch.mit.edu/>
- Para guardar los proyectos realizados en la Web, es necesario crearse una cuenta de usuario en la web de Scratch .
- Los proyectos Scratch se pueden exportar y acabar guardados en nuestro ordenador.
- Permite compartir nuestro proyecto a la comunidad de "Scratchers".
- Permite "embeber" el proyecto en una página web. [Ver cómo hacerlo](#).



## ¿Qué opción es la más recomendable?

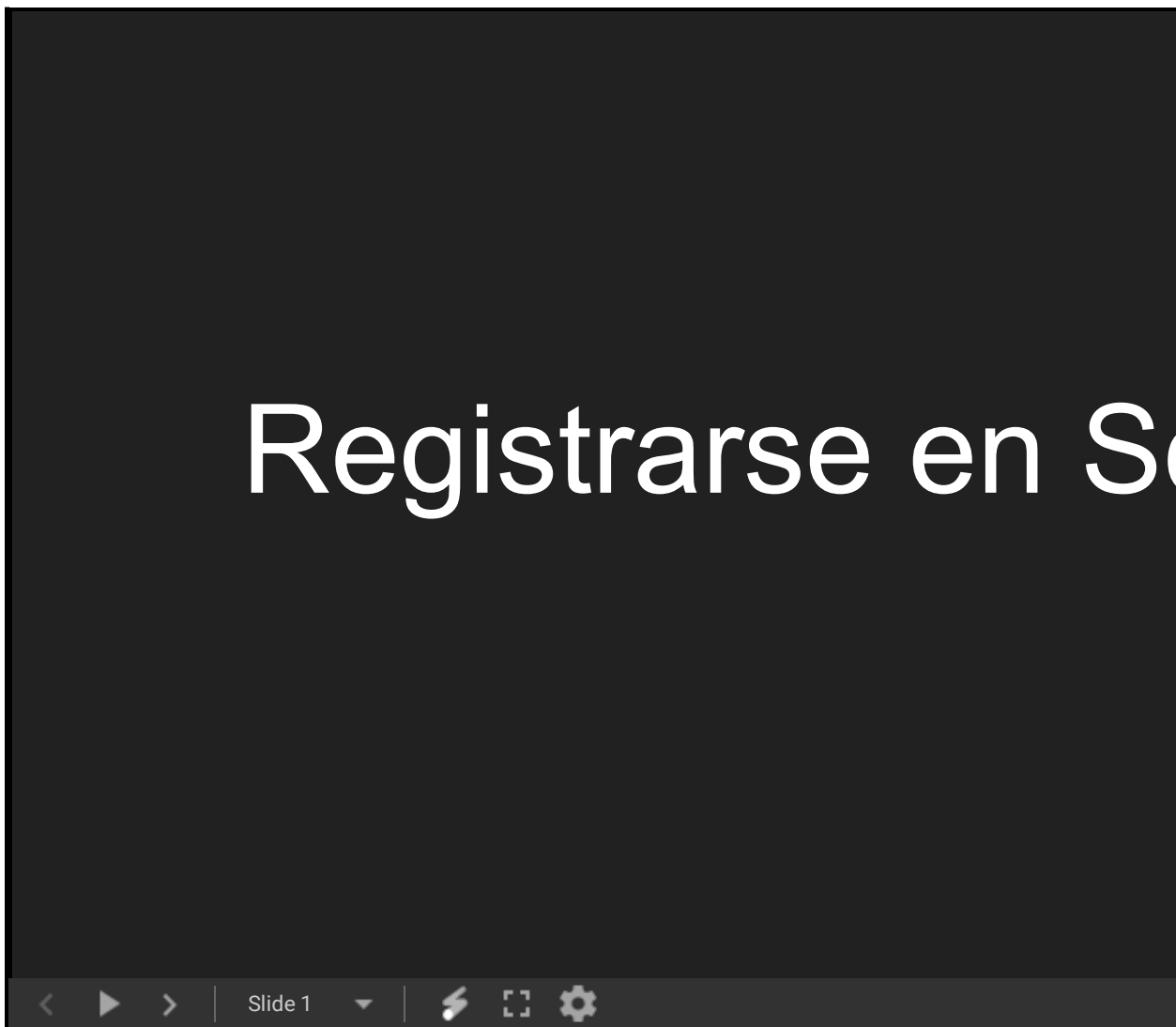
Lo mejor es la versión OnLine, sólo recomendamos utilizar la versión Offline si hay problemas de conexión de Internet.



## On-Line

Scratch on-line es la opción que recomendamos al menos que en tu centro tenga problemas de conexión de Internet.

Aquí tienes un [tutorial el FLASH](#) como registrarse o [en PDF](#) si lo ves mal **VERSIÓN ALUMNO**



**ATENCIÓN** si quieres registrarte como docente, entonces [consulta esta página](#)

(Si te registras como docente puedes crear como alumno, y además crearte tus clases y controlar a tus alumnos)

(Si te registras como alumno no puedes cambiar como docente)

# Off-Line

## Actividad: Instalar Scratch 2 Offline Editor

Entra en la web de Scratch <https://scratch.mit.edu/>

Al final de esa página está el enlace "Editor sin Conexión", haz clic en él. Si no lo encuentras, puedes ir directamente a la web de descarga del [Editor sin Conexión \(https://scratch.mit.edu/scratch2download/\)](https://scratch.mit.edu/scratch2download/):

1. Instala Adobe Air
2. Instalar Scratch Offline Editor
3. No es necesario que descargues los materiales de soporte del paso 3, pero son material complementario y de ayuda que te servirán para profundizar en Scratch
4. ¡Arranca Scratch en tu ordenador!

**Scratch 2 Offline Editor**

You can install the Scratch 2.0 editor to work on projects without an internet connection. This version will work on Mac, Windows, and some versions of Linux (32 bit).

**Note for Mac Users:** the latest version of Scratch 2.0 Offline requires Adobe Air 20. To upgrade to Adobe Air 20 manually, go [here](#).

Adobe AIR	Scratch Offline Editor	Support Materials
<p><b>1</b></p> <p>If you don't already have it, download and install the latest <a href="#">Adobe AIR</a></p> <p>Mac OS X - <a href="#">Download</a> ⬇</p> <p>Mac OS 10.5 &amp; Older - <a href="#">Download</a> ⬇</p> <p>Windows - <a href="#">Download</a> ⬇</p> <p>Linux - <a href="#">Download</a> ⬇</p>	<p><b>2</b></p> <p>Next download and install the Scratch 2.0 Offline Editor</p> <p>Mac OS X - <a href="#">Download</a> ⬇</p> <p>Mac OS 10.5 &amp; Older - <a href="#">Download</a> ⬇</p> <p>Windows - <a href="#">Download</a> ⬇</p> <p>Linux - <a href="#">Download</a> ⬇</p>	<p><b>3</b></p> <p>Need some help getting started? Here are some helpful resources.</p> <p>Starter Projects - <a href="#">Download</a> ⬇</p> <p>Getting Started Guide - <a href="#">Download</a> ⬇</p> <p>Scratch Cards - <a href="#">Download</a> ⬇</p>

También existe una versión para Android llamada ScratchJr [\[Ver\]](#). Es una versión reducida pensada para alumnos de infantil.

### Sobre la versión anterior Scratch 1.4:

- Los proyectos de Scratch 1.4 se guardan en archivos con extensión .sb
- Un proyecto realizado en Scratch 2.0 (extensión .sb2) puede convertirse en 1.4 (extensión .sb) usando [Retro Converter](#) pero algunos módulos propios de la versión 2.0 no se convertirán. [Ver página con información.](#)
- Un proyecto realizado en Scratch 1.4 se puede abrir en 2.0
- Un fichero .sb se puede convertir en un ejecutable exe de Windows o App en Mac o en un JAR Java ejecutable [Ver página.](#)

**Curiosamente no es posible compartir proyectos de S4A en la web de la comunidad Scratch, ya que ésto violaría los términos y condiciones de uso de Scratch. S4A es una modificación de Scratch para programar la plataforma hardware Arduino.**



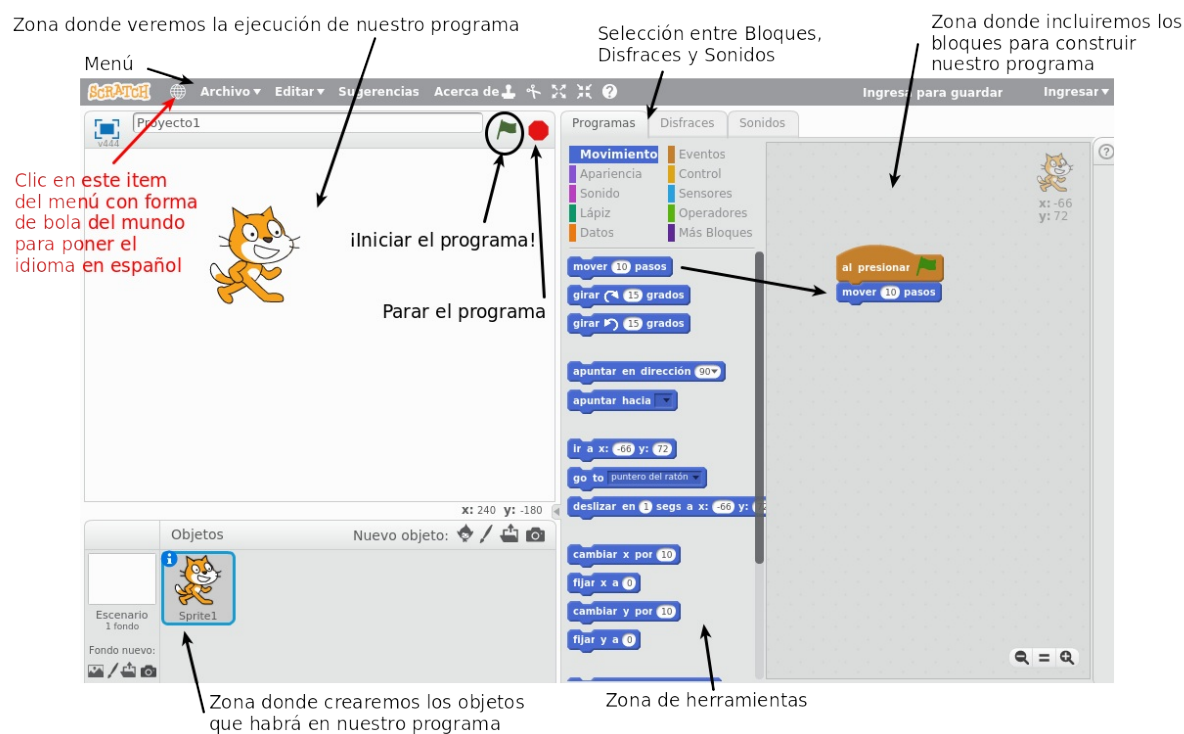
## Entorno Scratch

Una vez que hayas arrancado Scratch en tu ordenador, o que hayas entrado en la web online de Scratch, verás el entorno de trabajo de Scratch. En la siguiente imagen se muestran las diferentes zonas de trabajo de Scratch.

En el primer ítem del menú con forma de bola del mundo se puede cambiar el idioma a español.

Glosario:

- **SPRITE** Es el objeto sobre el cual programamos, en la figura de abajo es el gato, pero podemos incorporar más predefinidos de la biblioteca de Scratch, pintarlos con mapa de bits o vectorial, o incluso desde cámara web.
- **EVENTO** Suceso provocados por el usuario, los sprite o por el sistema, y sobre el cual queremos que el programa se ejecute o haga un comportamiento determinado, por eso en Scratch se dice que su programación es "orientada a eventos". Ejemplo de eventos provocados por ...
- **Sprite**: si el gato toca el borde, pues que haga...
- **Usuario** Si se apreta la tecla espacio que ...
- **Sistema**: Si hora actual = 12 que ....



## ¡Ya funciona!

### Actividad: ¡Ya funciona!

Vamos a hacer nuestro primer programa en Scratch:

1 Clic en la pestaña Programas - Eventos 2 Arrastra el bloque "Al presionar bandera":



3 Clic en en la pestaña Programas - Movimiento 4 Arrastra el bloque "mover 10 pasos" a la zona derecha y encájalo arrastrándolo justo debajo del bloque "Al presionar bandera":



Y ¡ENCAJALO!



5 Haz clic en la bandera verde para iniciar el programa:



¡Verás que el gato Scratch se mueve un poco hacia la derecha! Se mueve 10 pasos, correspondientes a 10 píxeles.

El programa termina sólo porque ha terminado de ejecutar todas las instrucciones. Por ello no hay que darle al botón Stop rojo (al lado de la bandera verde) para que finalice. El botón Stop servirá cuando nuestro programa tenga bloques de control que impliquen repetición continua de ejecución (esto se verá más adelante).

## 2 Empezando a manejar Scratch

En este bloque de contenidos vamos a empezar a usar Scratch. Se avanzará de forma progresiva, mostrando poco a poco las posibilidades que ofrece, e ¡incluso acabaremos consiguiendo realizar un juego de ordenador! ¿es fácil y posible? ¡sí!, ¡y se puede conseguir en muy poco tiempo!



## Bloques

Scratch dispone de un buen número de bloques. En el presente curso sólo vamos a estudiar y utilizar un subconjunto de ellos, serán lo suficientemente significativos para que aprendáis a utilizar Scratch de una forma completa y os permitirá tener una base para que vosotros mismos probéis el funcionamiento de otros bloques.

Si os gustaría tener información sobre todos los bloques impreso, os recomendamos esta guía de Eduteka donde aparece descrito lo que hace cada bloque: <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf#page=15>

## Actividad: Moviendo los bloques

Continuamos con el proyecto que ya funciona. Haz que en vez de 10 pasos, avance 50: Para ello haz clic en el número 10, borra con el teclado el 10 y escribe 50, y pruébalo de nuevo.

Manejando bloques: Observarás que los bloques en la zona del programa se pueden mover.

- Si mueves el bloque naranja "Al presionar bandera", se arrastrará también el bloque azul "mover 50 pasos".
- Si mueves el bloque azul "mover 50 pasos", se desencajará del bloque superior.

**Los bloques se quedan enlazados, y cuando arrastremos un bloque, con él se moverá todo lo que de él dependa, pero se desprenderá de sus bloques superiores.**

A continuación del bloque azul "mover 50 pasos", añade el bloque "esperar 1 segundo", que está en el los bloques de Programas - Control.

Añade también varios bloques azules más, de forma que queden intercalados, tal y como se muestra en la siguiente imagen, y pruébalo.



## Actividad: Borrando bloques

Para **quitar un bloque** (y todo lo que depende de él):

- Opción a: Arrástralo de nuevo a la zona de bloques de donde lo arrastraste.
- Opción b: También puedes darle con el "Botón derecho del ratón - Borrar".

Para **quitar un bloque que esté encajado en medio de otros bloques**: desencájalo de arriba, desencajale todo lo que tiene abajo, bórralo, y encaja los bloques superiores con los inferiores. En la siguiente imagen se muestra cómo poder borrar el bloque "mover 50 pasos" que está en el medio:





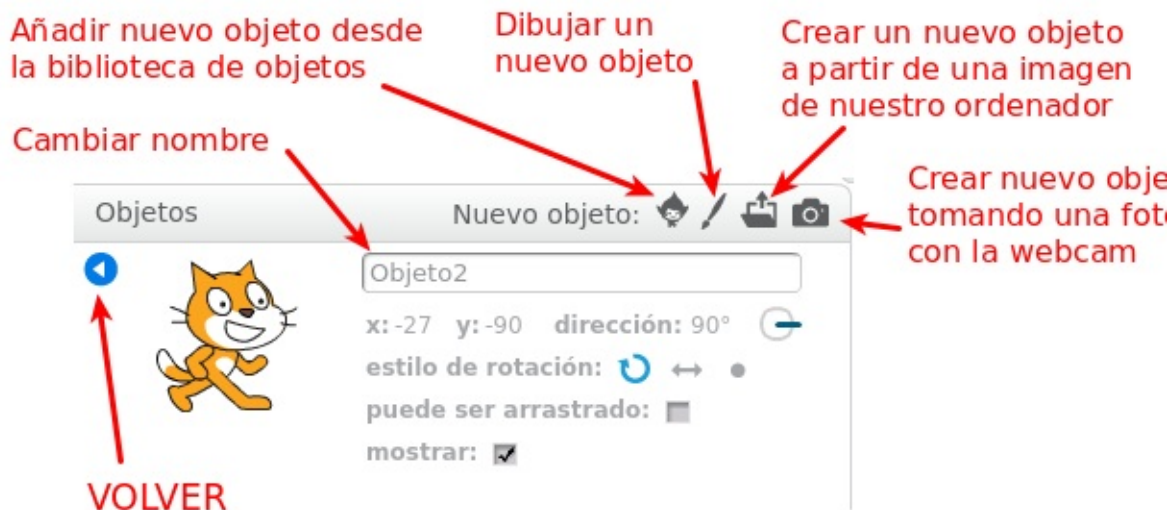
**DESHACER BORRADO:**

Si borras un bloque o un conjunto de bloques entero, puedes recuperar lo último que has borrado en el menú **Editar - Recuperar borrado**.

## Objetos, Disfraces y Fondos

### Actividad: OBJETOS

Un nuevo proyecto viene con un objeto: es un Gato. Se puede cambiar el nombre al objeto, dándole con el Botón derecho - Info y luego cambiándole el nombre en el recuadro.

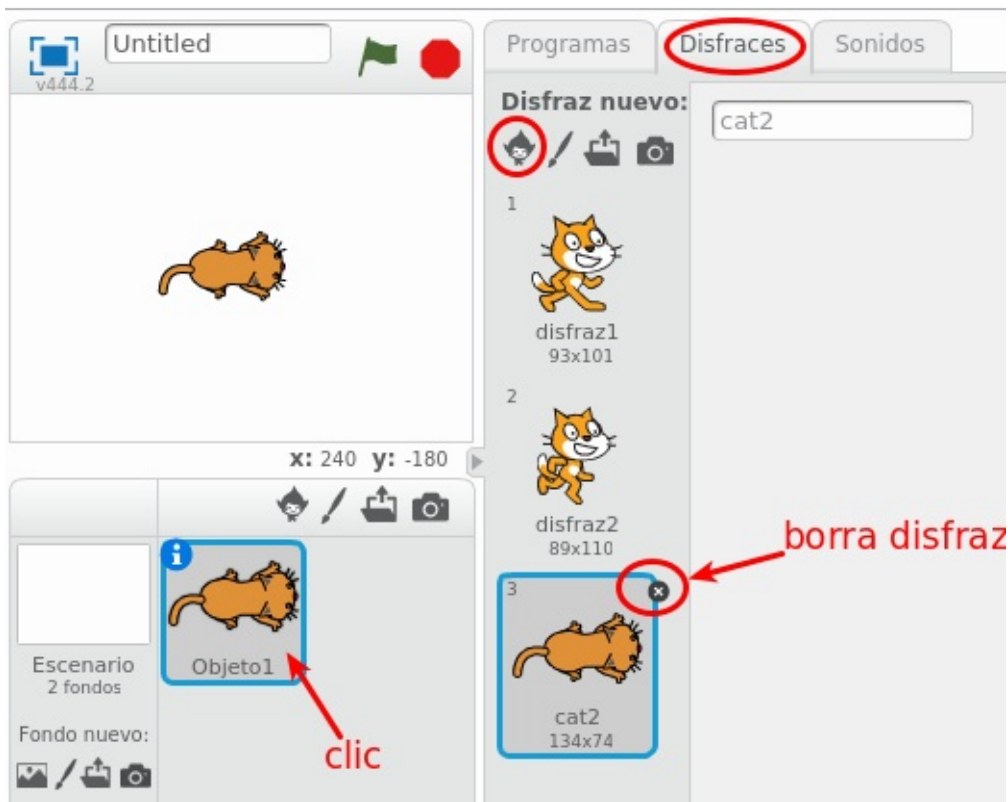


Añade

un nuevo objeto desde la biblioteca de objetos, en concreto el objeto "Dog1". Está en la categoría Animales.

### Actividad: DISFRACES

Cada objeto puede tener muchos disfraces. El Gato por defecto viene con 2 disfraces. Podemos añadir un nuevo disfraz a un objeto.



Añade un nuevo disfraz al

objeto Gato: Primero haz clic en el objeto Gato. luego añade el disfraz Cat2 al objeto Gato desde la biblioteca de disfraces.

También podemos borrar disfraces: Borra el disfraz que acabas de añadir.

## Actividad: FONDOS

Inicialmente el fondo es blanco. Podemos tener preparadas varias imágenes que podemos hacer que sean el fondo del programa.



Añade un fondo de la

biblioteca de fondos, por ejemplo el fondo "blue sky".



# Programas y Movimiento

## Actividad: Añadir otro programa

Si ya has trasteado mucho, cada vez que lo necesites, puedes volver a empezar un nuevo proyecto en blanco. Dale a menú **Archivo - Nuevo** para iniciar un nuevo proyecto, o también puedes ir borrando lo que ya no necesites.

Vamos a añadir que al presionar la tecla "a", el objeto se gire 90 grados:

1. Habrá que utilizar un bloque de Eventos, en concreto el bloque **"al presionar tecla espacio"** Pongo en cursiva espacio porque lo podremos cambiar. Haciendo clic en **"espacio"**, podremos cambiarlo por **"a"**.
2. Se necesita también el bloque de Movimiento **"mover 10 pasos"**. Haciendo clic en el **10**, podríamos ponerle 100 para que avance más.

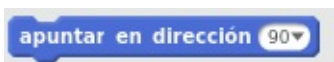


Observad que hay 2 programas en la zona derecha. Cada objeto puede tener varios programas, que funcionan de forma paralela.

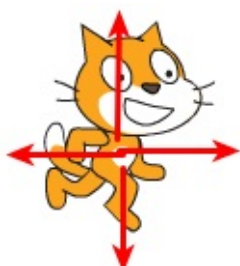
**IMPORTANTE:** Los programas se identifican por empezar con un bloque de "Eventos" cuya parte superior es ovalada.

## Caso práctico

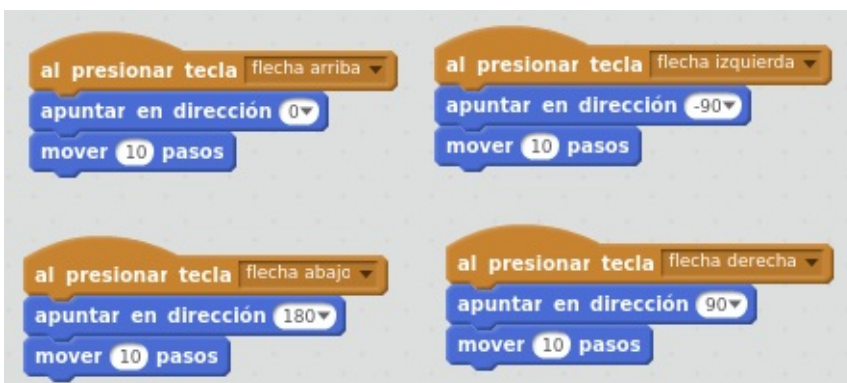
El objeto se mueve en la dirección en la que apunta. Hay un bloque que hace que el objeto apunte a otra dirección:



Haz que el objeto se mueva según las flechas de teclado.

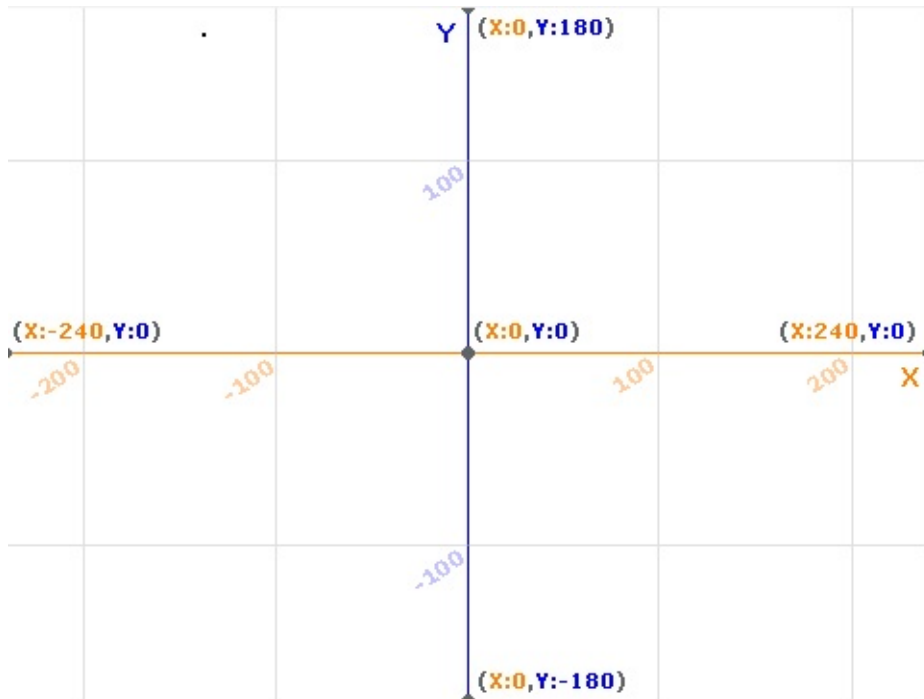


Solución



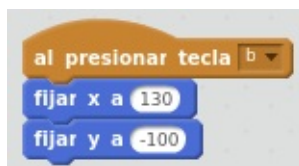
## Caso práctico

La posición de un objeto se controla con coordenadas X e Y. La posición de estas coordenadas en la pantalla es la siguiente:



Esta imagen es un Fondo que ya está creado en Scratch, y que puedes añadir a tu proyecto.

Añade este programa.



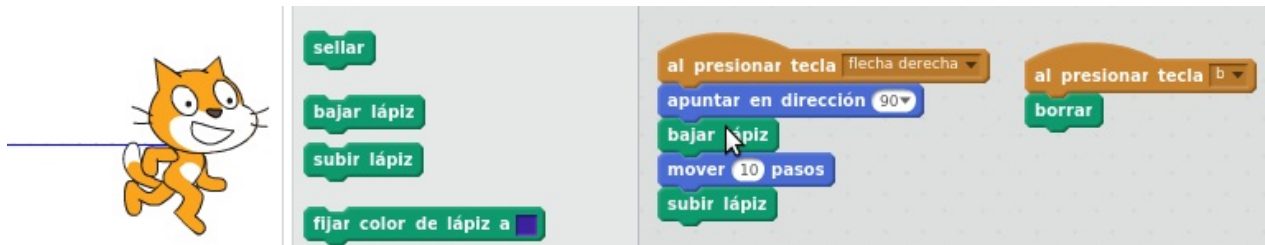
¿Qué hace este programa?

Solución

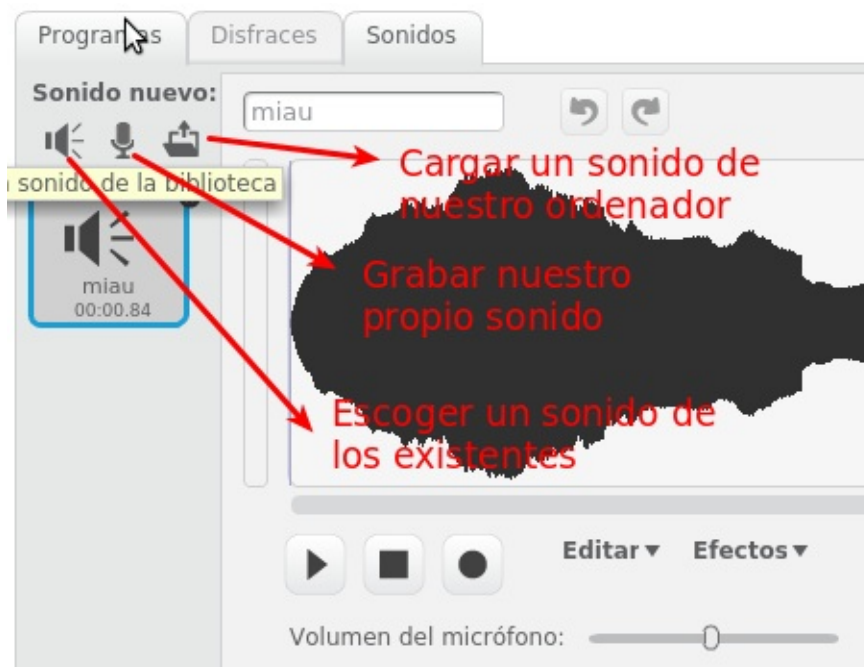
Este programa mueve el objeto a la zona inferior derecha de la pantalla cuando se presiona la tecla b.

## Lápiz y Sonido

Los bloques de **Lápiz** sirven para dejar una estela por donde el objeto se mueva. Para que deje la estela dibujada, es necesario añadir el bloque "bajar lápiz" previamente al movimiento. Cuando terminemos de pintar, habrá que poner el bloque de "subir lápiz" para que al volver a mover el objeto ya no pinte. Será posible pintar con distintos colores y grosores. El bloque "borrar" sirve para borrar todo lo pintado. Podemos poner un Programa que al darle a la tecla "b" borre todo. Puedes probar a poner estos bloques para que veas su efecto:



Los bloques de **Sonido** sirven para reproducir sonidos en algún momento de nuestro programa. Cada objeto viene con un sonido predeterminado. Pero nosotros mismos podemos grabar o añadir música que ya tengamos en nuestro ordenador:



## Caso práctico

Haz un programa que muestre el camino por donde se ha movido el gato. Añade que suene un "miau" después de haber pintado la línea.

Solución





## Control: condiciones

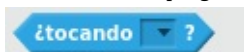
Las condiciones son un tipo de bloques que evalúan algo, y si es cierto, entonces se arrancan los bloques que tiene dentro anidado.

Añade estos bloques y pruébalos dándole a la bandera. (Los bloques verdes están en la pestaña Programas - Operadores). En estos bloques se comprueba que  $2+3=5$ , y como es cierto, entonces el gato dice "SUMA CORRECTA". Si cambias el 5 por cualquier otro valor, por ejemplo 6, la condición de  $2+3=6$  es falsa, y el gato dice "Hmmm... algo va mal".



## Caso práctico

Vamos a crear un programa que haga que el gato maulle cuando toque el borde del escenario. Para ello usaremos el bloque



Solución

Primero hay que crear los bloques que comprueben si estamos tocando el borde, y entonces maullar:



Una posible solución donde añadir la comprobación de tocar el borde, es añadirlo justo después de mover el gato en todas las direcciones.

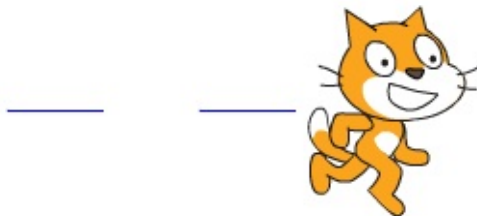




## Control: repeticiones

### Caso práctico

Para empezar a entender qué son las repeticiones, haz un programa que dibuje dos líneas discontinuas.



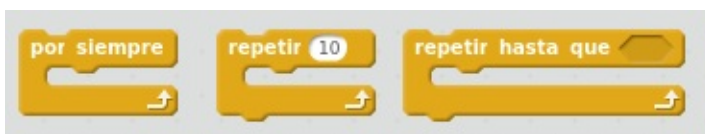
Solución



Puedes duplicar los bloques (botón derecho - duplicar) para no tener que construir dos veces lo mismo.

¿Y si hubiera que pintar muchas líneas?: es decir, repetir lo mismo muchas veces. Entonces hay que añadir un bloque de control para hacer repeticiones. Las repeticiones también se llaman "bucles". Es posible añadir 3 tipos de bucles:

- Se repiten constantemente, por siempre.
- Se repiten un número de veces concreto
- Se repiten hasta que se cumple una condición



### Caso práctico

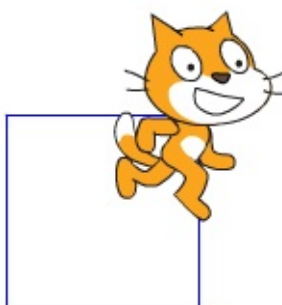
Haz un programa que dibuje 10 líneas discontinuas seguidas.

Solución



## Caso práctico

Haz un programa que dibuje un cuadrado.



Solución

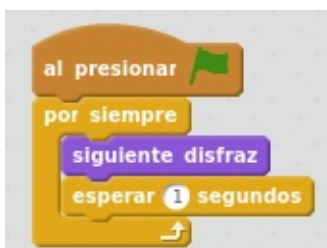
Hay dos soluciones posibles de dibujar el cuadrado utilizando el bloque de repeticiones. A continuación se muestran los dos programas posibles. Con que construyas uno de ellos ya tienes la solución. Los bloques de subir y bajar lápiz pueden estar dentro o fuera del bloque de repetición, en este ejemplo el resultado será el mismo.



## Caso práctico

Haz que el gato vaya cambiando de disfraz constantemente. Añade un retardo de 1 segundo para que el cambio de disfraz sea algo más lento.

Solución





## Mensajes

Podemos comunicar dos programas a través del envío de mensajes. Se puede enviar un mensaje desde un programa de un objeto a otro, para hacer en él alguna acción. Estos bloques se encuentran en "Eventos".



## Caso práctico

Vamos a hacer que cuando el gato toque al perro, enviemos un mensaje desde un programa del gato hasta un programa del perro. Cuando el perro reciba el mensaje, lo cambiaremos de posición, a una posición aleatoria.

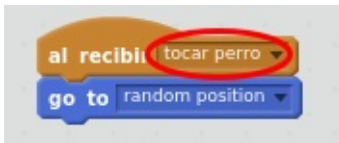
Solución

Al presionar Bandera, el objeto Gato envía un mensaje, lo recibimos en el objeto Perro, y hacemos que el perro se mueva a una posición aleatoria en pantalla.

Programa a añadir en el Gato:



Programa a añadir en el Perro:



## Variables

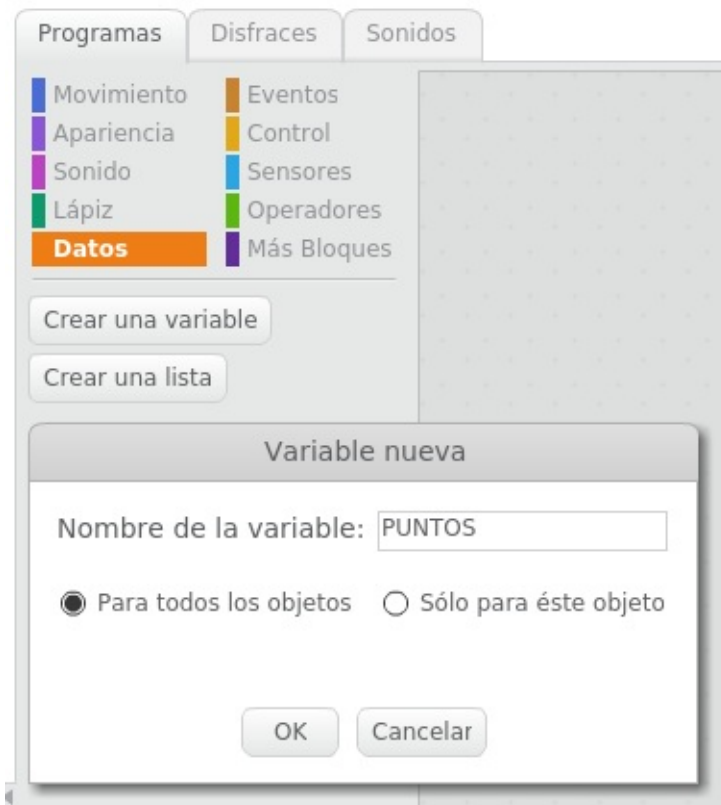
Las variables las podemos entender como un "nombre" que contiene un "valor". Podremos asignar un valor a la variable. Y podremos recuperar el valor de la variable. Al crear una variable, aparecen los bloques de control de las variables, en "Datos".



## Caso práctico

Crea una variable y ponle de nombre PUNTOS

Solución



- Haz que al iniciar el juego se ponga el valor de PUNTOS a 0
- Haz que al presionar la tecla "p" se incremente en 1 el valor de PUNTOS
- Haz que al hacer clic encima del Gato, el gato diga el valor de PUNTOS

Solución







## ¡El primer juego!

### Caso práctico: ¡Vamos a hacer el primer juego!

Con todo lo que hemos visto, ya tenemos todos los ingredientes para hacer un sencillo juego.

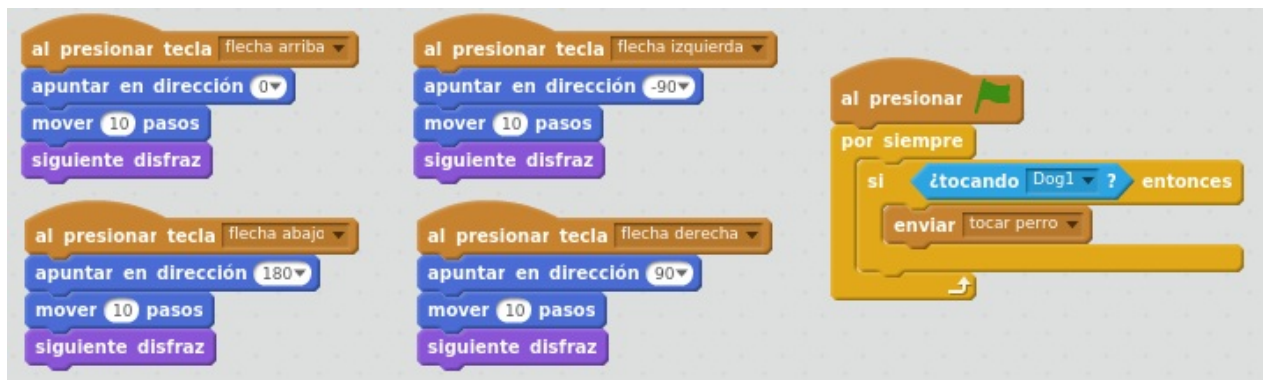
Moveremos el gato con las flechas del teclado, el gato cambiará de disfraz para parecer que va andando.

Hacer que cuando el gato toque al perro, el perro se mueva a una posición aleatoria y ladre.

Solución

En una actividad anterior hemos visto que el bloque "Si....entonces" lo podemos añadir por cuatro veces después de presionar cada tecla. En esta solución se muestra cómo poner estos bloques sólo una vez en un programa independiente dentro de la programación del Gato, y es necesario añadir el bloque "por siempre" para que se esté constantemente comprobando si el gato toca al perro.

**Programación del Gato:**



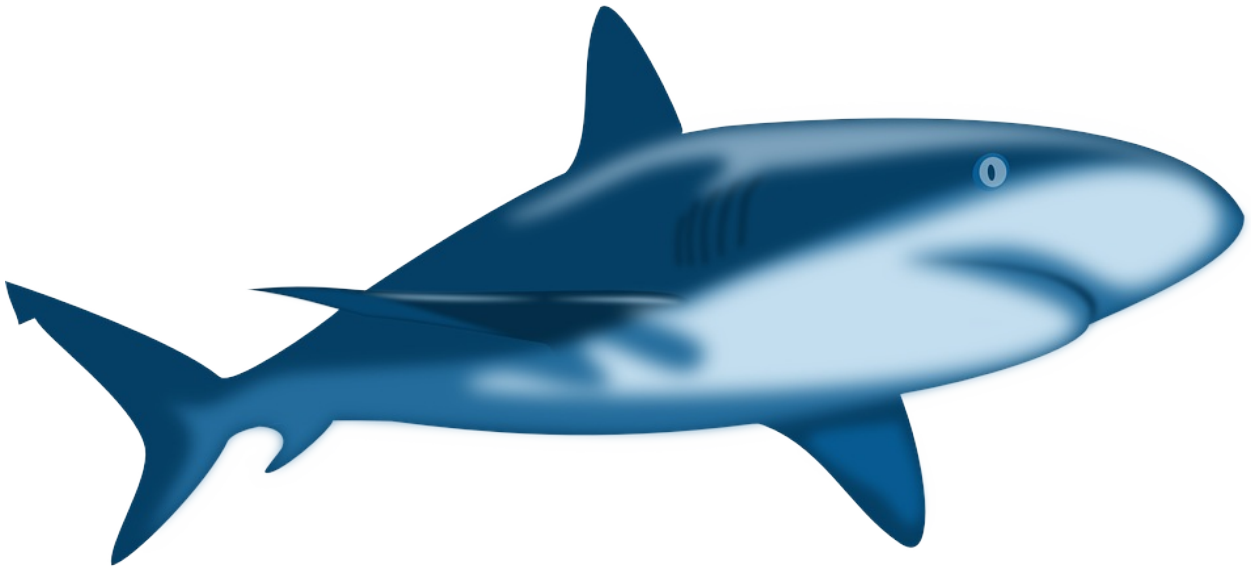
**Programación del Perro:**



## Módulo 3 Proyecto

En este módulo vamos a trabajar un poco más las instrucciones aprendidas, la mejor forma es haciendo "Gamificación" es decir "Aprendiendo jugando"

Y los niños les gusta la acción... así que vamos a ser un poco niños ;)



Pixabay.com CCO

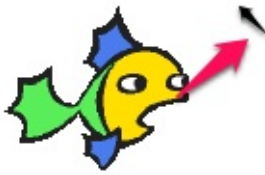
## Movimiento pez

### Vamos allá primer paso ....

El primer paso que vamos a hacer es :

**Realizar un programa para conseguir que el "pez persiga al ratón"**

1. Insertar un SPRITE predeterminado que se llama "fish2" (en biblioteca - sección animales)
2. Crear un programa que siempre persiga la posición del ratón



Guardalo que lo utilizaremos más adelante

Solución



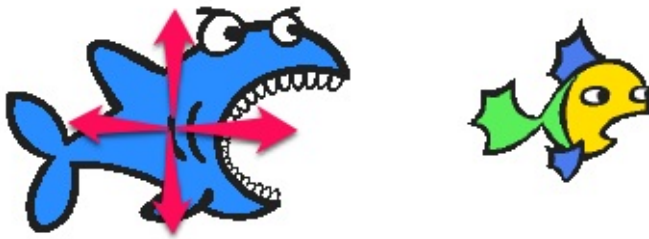
# Movimiento tiburón

## Segundo paso

Hora vamos a añadir un

**Un tiburón que se mueva según apretemos las teclas de flechas del teclado**

1. En el mismo programa, añade el sprite de la biblioteca Shark
2. Haz un programa al sprite Sharck que se mueva según las flechas del teclado (pista: Ya lo vimos ;)



Guardarlo en el mismo fichero!!!

Solución



# Aparencia

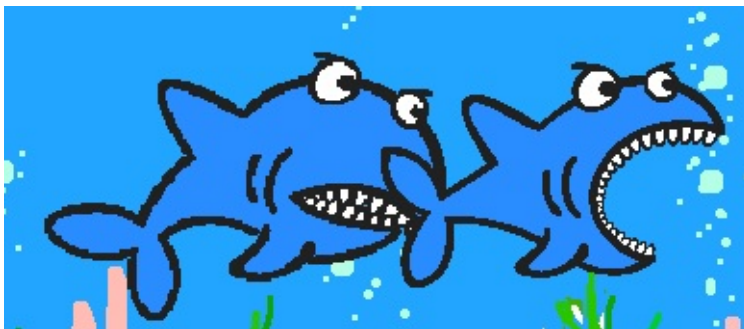
## Tercer paso

Ahora lo que vamos a hacer es que ...

**El tiburón se desplace con sensación de estar "masticando" y poner un escenario adecuado**

Es decir, realizar un programa que:

1. El sprite shark vaya cambiando de disfraz sin parar. Como hemos utilizado el sprite shark predefinido de la biblioteca, tiene ya los disfraces de abrir y cerrar la boca.
2. Poner el escenario adecuado, hay uno predefinido de fondo marino.



Solución



# Sonido

## Cuarto paso

Algo de música ¿no? las cosas con un poco de sal saben mejor ;)

### Poner una música de inicio al programa

Por ejemplo Xilo2 o la que tu quieras



### Solución

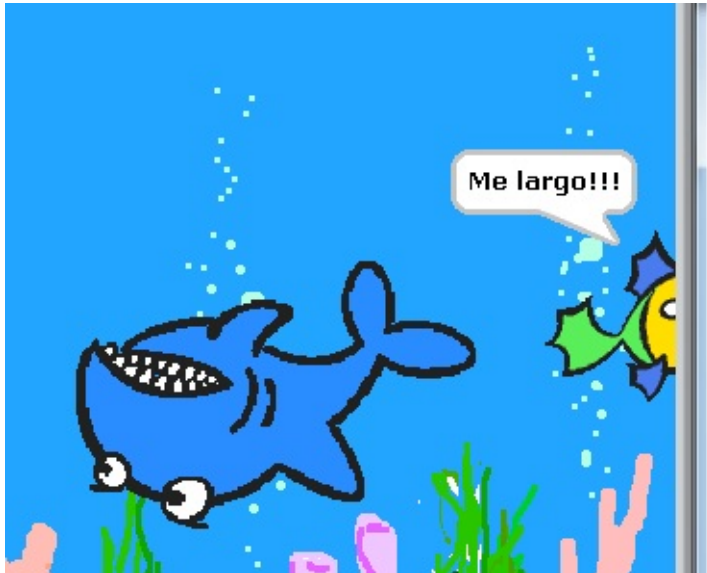


# Me largo!!!

## Quinto paso

El pez pequeño si toca un borde que diga que se larga!

Por ejemplo el mensaje puede ser "Me largo !" durante 0.2 seg



Solución



# Cobarde !!!

## Sexto paso

El tiburón diga "Cobarde" si el pez pequeño se larga

Es decir, si el pez pequeño se toca el borde (actividad anterior), tiene que enviar un m.... y si el tiburón recibe el m.... entonces que diga Cobarde!! durante también 0.2seg



Solución

Exacto! lo de m.... quería decir **mensaje**.

El pez pequeño envía un mensaje:



y el tiburón recibe el mensaje y contesta:





## Dos jugadores

### Ya toca jugar con dos jugadores Paso final, la puntuación

Como la vida misma: Si no hay premio o cuantificamos, no hay animación !

#### Idear un sistema de puntuación en el juego

Por ejemplo podría ser este:

- Si el pez grande toca al pequeño:
  - el pez grande gana un punto
  - suena un "zoop"
- Si el pez pequeño toca el borde:
  - el pez grande gana otro punto

Sí, ya sé que es un poco injusto que el pequeño no gane nada, pero así es la vida...

Se recomienda redimensionar los peces para que sean pequeños



Solución

Creamos una variable que lleve la puntuación, aquí lo hemos llamado "score"

Y el reto es dónde poner la instrucción que va incrementando la puntuación "cambiar score a 1"

En el programa del pez pequeño:



Programa del tiburón



[Descarga del programa](#)

## Consejos

### ¿Ya sabes lo básico de Scratch?

Ponte a prueba:



[Video link](#)

2017 por [CATEDU](#) (Javier Quintana Peiró).

Cualquier observación o detección de error por favor aquí [sopORTE.catedu.es](mailto:sopORTE.catedu.es)

Los contenidos se distribuye bajo licencia Creative Commons tipo BY-NC-SA.



# GOBIERNO DE ARAGON

Departamento de Educación,  
Cultura y Deporte

**CATEDU**



CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN

