

Guía Didáctica para Profesores



SCRATCH





Esta guía didáctica ha sido elaborada por Isuri Sormenezko Zerbitzuak
www.isuriarte.com

Para la realización de esta guía didáctica se ha partido de varios documentos y páginas web:

- http://edu.mec.gub.uy/banco%20de%20recursos/webscratch/mini_guia.htm
- <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>
- <http://s3.amazonaws.com/jef.mindtouch.com/10033948/1302/0?AWSAccessKeyId=1TDEJCXAPFCDHW56MSG2&Signature=vJ6D0G/d0bkz%2bJop8zTLYbQ6p%2bI%3d&Expires=1285067416>
- <http://fcp.unach.mx/manuales/download/Manual%20de%20scratch.pdf>
- <http://www.scratchbydsigno.com.ar/>

En la elaboración de esta guía, se han utilizado partes íntegras de los documentos anteriormente señalados, combinándolas con textos e imágenes nuevas, con la intención de mejorar y aunar en una guía más completa, los materiales publicados con anterioridad.

Scratch es un proyecto de Lifelong Kindergarten Group en el MIT Media Lab
scratch.mit.edu
LLK.media.mit.edu
media.mit.edu

Este documento está registrado bajo una licencia Creative Commons 3.0 España.
Siéntete libre de usarlo y manipularlo para adaptarlo según tus necesidades.



<http://creativecommons.org/licenses/by-nc-sa/3.0/es/legalcode.es>



¿Qué es Scratch?

Scratch es un nuevo entorno de programación visual y multimedia basado en [Squeak](#). Está pensado y diseñado para ser usado por lo niños y les permite crear sus propios videojuegos, historias animadas y arte interactivo. Por último podrán publicar sus creaciones para que todo el mundo pueda verlas.

Scratch utiliza la metáfora de “piezas encajables”, al igual que lo hacen Logo, LEGO/Logo o los Etoys de Squeak, para animar objetos que se encuentran en la pantalla, con un uso muy sencillo e intuitivo.

Scratch se utiliza desde un “entorno de desarrollo” que muestra de un solo golpe de vista todos los elementos necesarios: escenario, objetos y elementos del lenguaje. Permite tener tantos escenarios y objetos como se desee, utilizando aquellos que ya están disponibles con la instalación estándar de la herramienta, o bien creando otros nuevos. Este es un factor motivacional más a la hora de trabajar con el alumnado desde edades tempranas.

Prácticamente todo se lleva a cabo arrastrando y soltando elementos con el ratón, y modificando con el teclado únicamente los valores numéricos y textos.

¿Dónde descargarlo?

Ir a la página web de Scratch: <http://scratch.mit.edu/>

En la zona de “descargas” de la página web, se puede elegir entre las tres versiones disponibles: Windows, Mac y Linux. Para cada una de ellas se especifican los requisitos necesarios.

La página Web de Scratch tiene muchos otros recursos para ayudar a aprender Scratch: Tutoriales en video, Tarjetas de Scratch, y Preguntas frecuentes (FAQ). Consulte <http://scratch.mit.edu/howto>

¿Para qué sirve?

Trabajar con Scratch ayuda a comprender fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa, como son:

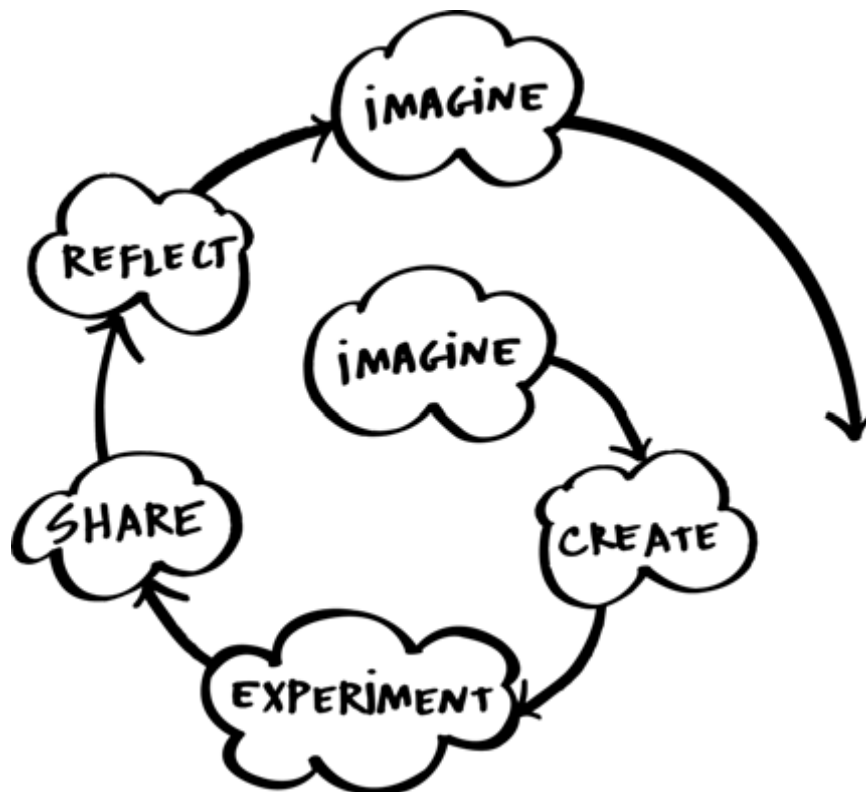
- Los procesos interactivos (bucles)
- Los criterios condicionales (si, entonces, si-no)



- Las coordenadas en un plano
- Las variables
- Etc.

Estos conceptos se aprenden dentro de un contexto significativo y motivador. Por ejemplo, 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.

Con Scratch las alumn@s trabajan en sus propios proyectos, de este modo aprenden sobre el **proceso de diseño**. Partiendo de una idea, tienen que crear el prototipo funcional, es decir, un modelo, y experimentar las soluciones que consideren necesarias para conseguir un producto final. Cuando estas ideas no funcionan tendrán 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 así, indefinidamente.



Estos procesos de concepción de proyectos y su construcción, desarrollan las competencias necesarias para llegar a tener:

- Un pensamiento creativo.
- Un pensamiento lógico.
- Un desarrollo de ideas, desde su concepción inicial hasta el proyecto acabado.
- Una comunicación clara.
- Un análisis sistemático.



- Capacidad de colaboración.
- Una reflexión interactiva.
- Etc.

De la misma forma que es necesario aprender no solamente a leer sino también a escribir, con la informática, l@s alumn@s no solamente deben aprender a utilizar e interactuar con el ordenador, sino también deben aprender a crear con esta herramienta.

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

Scratch y las competencias básicas

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.

Tratamiento de la información y competencia digital:

Trabajar con Scratch permite a l@s alumn@s 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.

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.

Desarrollo de las capacidades creativas y la curiosidad intelectual:

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

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.



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.

Apariencia del programa

ESCENARIO: Es la zona más grande y blanca, donde está la mascota de Scratch cuando se abre el programa. Es donde se desarrollara la acción.

MODO PRESENTACIÓN: Está en la esquina superior derecha de la pantalla. Da tres opciones de visualización del escenario.

BOTONES NUEVOS PERSONAJES/SPRITES: Justo debajo del escenario hay tres botones. Van a permitir buscar o incluso crear nuevos actores para la acción.

LISTA DE SPRITES: Debajo de los botones anteriores, aparecerán las miniaturas de los personajes que vayan a actuar. Hacer clic para seleccionar y editar un personaje.

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.

PALETA DE BLOQUES: Situada a la izquierda. Es la caja de instrucciones para que los actores hagan cosas. Son los bloques para crear los scripts o programas. Hay 8 paletas diferentes de piezas, para distintas acciones, cada una de un color: Movimiento, apariencia, sonido, lápiz, control, sensores, números y variables.

AREA DE SCRIPTS (guiones): Entre la paleta de bloques y el escenario, queda otro área muy importante, es área de scripts. Aquí se irán arrastrando los bloques para construir los guiones o programas.

PESTAÑAS: Justo encima del área de scripts tenemos tres pestañas que nos permiten añadir guiones, disfraces o sonidos al guión que construimos.



ESTILO DE ROTACIÓN: Está encima de las pestañas y un poquito a la izquierda. Nos permite determinar la rotación de los sprites.

Una vez explicadas cada una de las partes que forman el programa y a continuación se pueden encontrar dos pantallazos. En el primero se muestra la pantalla principal del programa y se detallan cada una de las partes. El segundo, se trata de cada una de las paletas de bloques desplegada.

Paletas de bloques

Menú principal: opciones para abrir, guardar o crear proyectos

Bandera verde: inicia el proyecto
Botón rojo: detiene el proyecto

Modo presentación: formas de visualización de pantalla.

Escenario: todo ocurre aquí

Coordenadas X e Y: indican la posición del ratón. Útil para planificar desplazamientos o detectar fallos.

Espacio con triple propósito: El propósito lo define cada una de las tres pestañas que aparecen en la parte superior. **SCRIPTS** es para disponer bloques programando acciones. **DISFRACES** son los distintos "trajes" que puede tener el personaje y **SONIDOS** que el personaje tenga asignados. Se pueden grabar nuevos sonidos.

Personajes/sprites: Aquí puedes dibujar, importar u obtener sprites "sorpresa" para tus proyectos. A cada sprite le acompaña un breve resumen de cuantos trajes (Costumes) y cuantos guiones (Scripts) posee.

Propiedades del escenario: Se puede cambiar el fondo y tener varios fondos a la espera de ser mostrados cuando sea preciso. Imagina una obra teatral con múltiples escenografías.

Área de bloques pertenecientes a una paleta: Si elijo la paleta de bloques correspondientes a CONTROL, solo aparecen los bloques que tengan vínculo con esa paleta.

Bloques

Movimiento **Control**
Apariencia **Sensores**
Sonidos **Números**
Lápiz **Variables**

mover 10 pasos
 girar 15 grados
 girar 15 grados
 apuntar en dirección 90
 apuntar hacia
 ir a x: -139 y: -58
 ir a
 deslizarse 1 seg a x: -139 y: -58

cambiar x por 10
 fijar x a 0
 cambiar y por 10
 fijar y a 0

rebotar si está tocando un bord

posición x
 posición y
 dirección

+ mod
 - abs
 * redondear
 /

número al azar entre 1 y 10

< y
 = o
 > no

cambiar el disfraz a costumel...
 siguiente disfraz
 decir Hello! por 2 segundos
 decir Hello!
 pensar Hmm... por 2 segundos
 pensar Hmm...

cambiar efecto color por 25
 fijar efecto color a 0
 quitar efectos gráficos

cambiar tamaño por 10
 fijar tamaño a 100 %
 tamaño

mostrar
 esconder

enviar al frente
 enviar atrás 1 capas

Nueva variable
 Borrar variable

cambiar ALFAJORES por 1
 fijar ALFAJORES a 0
 ALFAJORES

tocar sonido pop
 tocar sonido pop y esperar
 detener todos los sonidos

tocar tambor 48 durante 0.25
 tocar nota 60 durante 0.5 se
 fijar instrumento a 1

borrar
 bajar lápiz
 subir lápiz

fijar color de lápiz a
 cambiar color de lápiz por 10
 fijar color de lápiz a 0

cambiar intensidad de lápiz por
 fijar intensidad de lápiz a 50

cambiar tamaño de lápiz por 1
 fijar tamaño de lápiz a 1

sellar

Catálogo los bloques disponibles para ensamblar

al presionar
 al presionar tecla space
 al presionar Sprite2

esperar 1 segundos

por siempre

repetir 10

enviar a todos
 enviar a todos y esperar

al recibir

por siempre si
 si
 si
 si no

esperar hasta
 repetir hasta

SCRATCH

Fíjate cómo en la familia de bloques NUMEROS hay algunos bloques con extremos angulosos. Esos bloques especiales de comparación o de operaciones lógicas, solo encajan en el interior de algunos bloques de control

CONCEPTOS ESPECÍFICOS DE PROGRAMACIÓN

En este apartado 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.

CONCEPTO	EXPLICACIÓN	EJEMPLO
Secuencia	Para crear un programa en <i>Scratch</i> , se necesita pensar sistemáticamente el orden de los pasos a seguir.	
Iteración (looping)	Los bloques 'Por siempre' y 'Repetir' pueden ser usados para la iteración (repetir una serie de instrucciones).	
Condicionales	Los bloques 'Si...' y 'Si...no...' dan cuenta de una condición.	
Números aleatorios	El bloque 'Número al azar entre... y...' selecciona números enteros aleatorios dentro de un rango dado.	

<p>Variables</p>	<p>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.</p>	
<p>Listas</p>	<p>Los bloques 'Al presionar tecla' o 'Al presionar objeto' (o también un <i>sprite</i>) son ejemplos de la gestión de un evento que responde a eventos gatillados por el usuario u otra parte del programa.</p>	
<p>Hebras Temáticas (ejecución paralela)</p>	<p>Arrancando dos pilas de bloques al mismo tiempo, se crean dos hebras temáticas independientes que se ejecutan en forma paralela.</p>	
<p>Coordinación y sincronización</p>	<p>Los bloques 'Enviar a todos' y 'Al recibir' pueden coordinar las acciones de múltiples <i>sprites</i>. 'Enviar a todos y esperar' nos permite sincronizar acciones.</p>	<p>Por ejemplo, el Sprite 1 envía el mensaje "Ganador" cuando esta condición es recibida:</p> <p>Este script correspondiente al Sprite 2 es gatillado cuando el mensaje es recibido:</p>

<p>Entrada de teclado</p>	<p>El bloque 'Preguntar y esperar' invita al usuario a tipear en el teclado. 'Respuesta' almacena la entrada tecleada.</p>	
<p>Lógica Booleana</p>	<p>Los bloques '...y...', '...o...', 'no...' son ejemplos de lógica booleana.</p>	
<p>Interacción dinámica</p>	<p>Los bloques 'x del ratón', 'y del ratón' y 'Volumen del sonido' pueden ser usados como entradas dinámicas para interacción en tiempo real.</p>	
<p>Diseño de interfase de usuario</p>	<p>Puedes diseñar interfaces de usuario interactivas en Scratch – por ejemplo, usando <i>sprites clickeables</i> para crear botones.</p>	

Si se requiere una mayor precisión o explicación en el uso del programa, los profesores pueden consultar la *Guía de referencia de Scratch*, un amplio documento explicativo sobre la utilización del programa, elaborado por Eduteka. <http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>

PROPUESTA DE TRABAJO

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

Otra de las características que convierten a Scratch en una **herramienta colaborativa** interesante es la **posibilidad de 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** oficial de Scratch. El propio programa se encarga de comprimir todo el material utilizado y enviarlo a la web.

También se puede fomentar **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** de los mismos.

Educación Primaria:

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 complejidad**. Se puede comenzar a utilizar la herramienta cuando se desee, incluso con alumnado muy joven, haciendo uso únicamente de **proyectos ya realizados**, los cuales pueden **visualizar o 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 historias**. Se pueden ubicar 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 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>

Educación Secundaria:

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

Se puede comenzar a **asignar eventos a los personajes**, a **utilizar el teclado y el ratón para manejar los objetos**, y sobre todo a **asignar acciones que ocurren cuando varios objetos interactúan entre sí**. Así 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.

MÁS INFORMACIÓN Y RECURSOS

<http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf> *Guía de referencia de Scratch*: amplio documento explicativo sobre la utilización del programa, para profesores. (castellano)

<http://s3.amazonaws.com/jef.mindtouch.com/10033948/107/0?AWSAccessKeyId=1TDEJ CXAPFCDHW56MSG2&Signature=cqjKM%2bZC4tSme04n5Q2LfjVvk85A%3d&Expires=1285067443> Guía de inicio *Scratch* oficial. Muy útil para utilizar recurso didáctico, documento del alumno, explicaciones paso a paso. (castellano)

<http://scratch.wik.is/Languages> Página Web con listado de recursos y guías de *Scratch* en diferentes idiomas.

<http://scratched.media.mit.edu/resources> Página Web de recursos para profesores. (inglés)

<http://www.scratchbydsigno.com.ar/index.html> Página Web sobre Scratch. (castellano)

<http://aulavirtual2.educa.madrid.org/course/view.php?id=665> *Multimedia Scratch: crear como un niño*. Ejemplo de posible diseño de curso en *Moodle*. (castellano)

<http://scratch.mit.edu/tags/view/iktmagisteritza> recursos didácticos creados con *Scratch*, por los alumnos de magisterio de la Universidad del País Vasco. (euskara)