



Una herramienta lúdica de iniciación a la programación

# SCRATCH

El famoso Laboratorio Multimedia del Instituto de Tecnología de Massachussets y la Universidad de California, los Ángeles, han venido desarrollando, durante los tres últimos años, un interesante proyecto educativo destinado a elaborar *una divertida y atractiva herramienta de introducción a la programación*. A primeros de enero de este año tuvo lugar el lanzamiento oficial de Scratch [1] para Windows y Mac, y se prevé que a finales de este mismo año esté disponible la versión para Linux. **POR MÁXIMO PRUDENCIO**

**N**o obstante, por tratarse de una evolución de Squeak, podremos ejecutar la imagen de Scratch en una plataforma Linux siempre que tengamos instalada la máquina virtual de Squeak. Scratch es un entorno Squeak que funciona, con algunas limitaciones, sobre cualquier máquina virtual Smalltalk/Squeak. Por tanto, lo único que necesitamos es hacernos con la imagen de Scratch y enchufarla a nuestra máquina virtual [2].

## ¿Qué es Scratch?

Scratch es, para sus creadores, un medio de expresión mediante el cual los jóvenes y menos jóvenes pueden expresar sus ideas y responde a la pretensión de proporcionar una herramienta que facilite el uso de los ordenadores de forma creativa, superando el modelo de formación tradicional, que viene utilizando las nuevas tecnologías para reproducir prácticas educativas obsoletas.

Utilizar herramientas de programación como medio para alcanzar una mejor comprensión de las nuevas tecnologías no es nada nuevo, como tampoco lo es la valoración educativa de los algoritmos de programación. Lo novedoso es la sencillez del entorno lúdico desarrollado por este equipo de investigación del MIT liderado por el profesor Mitchel Resnick.

En cuanto herramienta de trabajo, Scratch es un proyecto de desarrollo



Figura 1: Rutinas informáticas en Scratch.

cerrado y código abierto. Es decir, al contrario de los proyectos convencionales de código abierto, el equipo de desarrollo no persigue la contribución de la comunidad de usuarios, sino que se reserva la elaboración de la versión estándar. Por otra parte, es un proyecto de código abierto porque el equipo de desarrollo espera liberar el código fuente para mediados de este año, de modo que pueda experimen-

tarse con extensiones y modificaciones del programa. No obstante, la versión final para Linux diferirá visualmente y será incompatible con la versión estándar para evitar la confusión entre ambas versiones.

### Los Orígenes

Scratch reconoce la aportación del micromundo de Logo, los e-toys de Squeak y LogoBlocks como sus precedentes o fuentes en las que se ha inspirado. El lenguaje de programación de Scratch, por un lado, se basa en Logo, en especial en sus primitivas, y presenta un entorno en el que múltiples objetos pueden evolucionar e interactuar. Por otro lado, como en Squeak -lenguaje en el que está desarrollado-

modo de trabajo consiste en arrastrar y soltar bloques en lugar de escribirlos. Y, por último, como LogoBlocks, CricketBlocks o PicoBlocks, utiliza bloques autoencajables que sólo ajustan si son sintácticamente correctos, permitiendo al usuario centrar su atención en los algoritmos lógicos de programación, en lugar de perder tiempo intentando desentrañar el esotérico código lingüístico de los tradicionales lenguajes de programación.

### Introducción a la Programación

A pesar de haber sido ideado como una sencilla herramienta para jóvenes, Scratch ya ha demostrado ser un instrumento valioso de introducción a la pro-



Figura 2: Metáfora de los bloques de construcción.

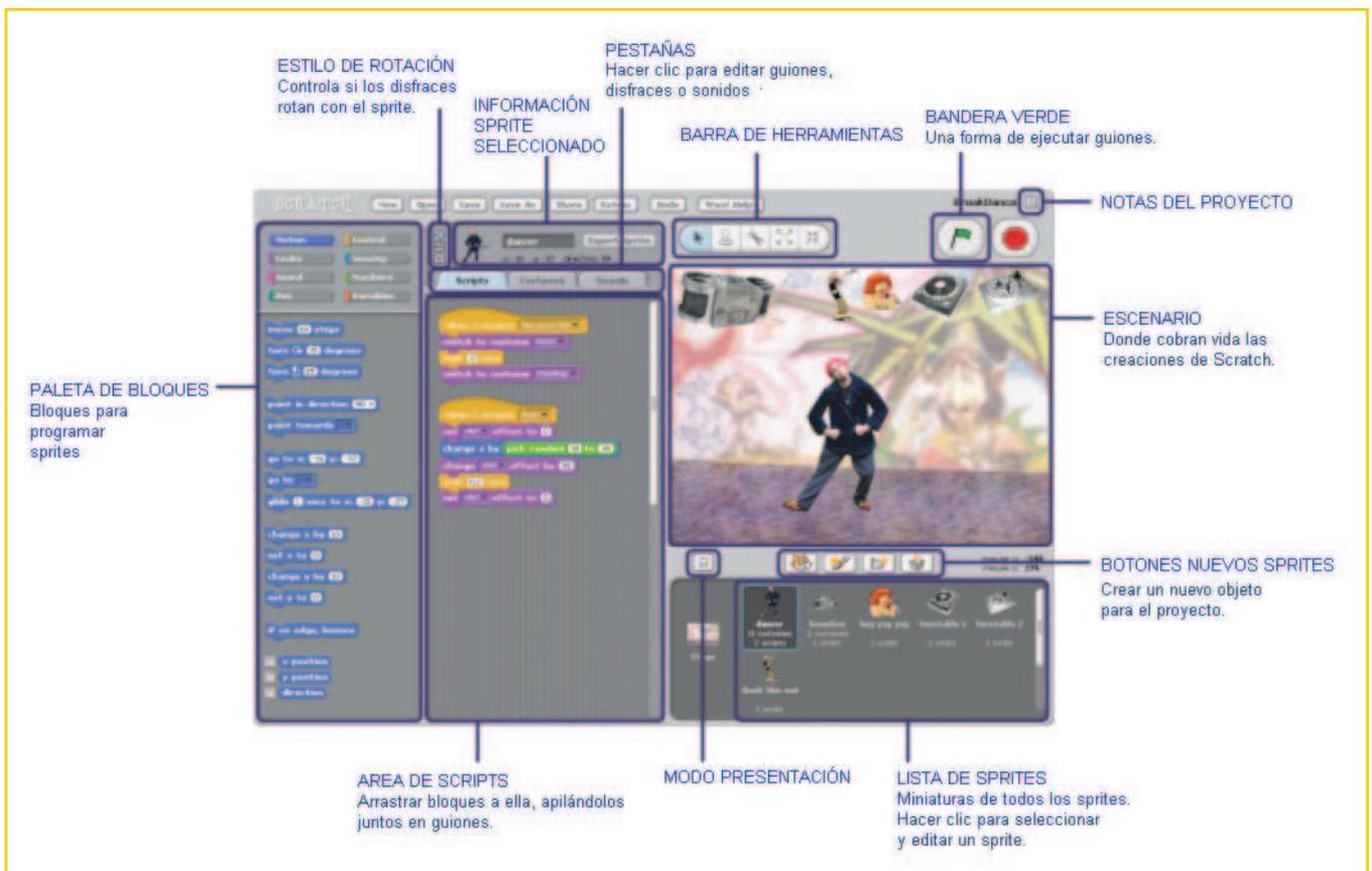


Figura 3: El interfaz de Scratch.

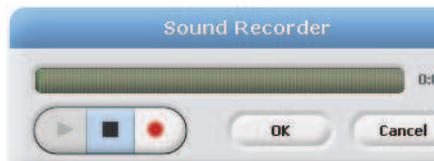


Figura 4: La grabadora de sonidos.

gramación en cursos desarrollados en Harvard por David Malan y Henri Leitner [3], en los que se ha comprobado la utilidad de usar Scratch como *calentamiento previo* para introducir Java en clases de programación, facilitando la transferencia de conceptos desde Scratch a otros lenguajes.

Al trabajar con proyectos de Scratch, los jóvenes tienen oportunidades para aprender conceptos informáticos importantes tales como iteración, control de flujo, condicionales, variables, tipos de datos, eventos y procesos. Scratch se ha usado para introducir estos conceptos a estudiantes de diferentes edades, desde la escuela primaria hasta el instituto, comprobándose cómo éstos migran con facilidad a los lenguajes tradicionales basados en texto después de haberse introducido en la programación mediante Scratch.

Scratch se ha optimizado para que sea fácil de escribir y, en menor medida, fácil de entender para programas relativamente pequeños en un dominio especializado, exactamente lo mismo que se podría decir acerca de lenguajes como Perl, Tk, Awk o APL. Ninguno de ellos es particularmente eficiente o se encuentra disponible para construir grandes aplicaciones. No obs-

tante, en opinión de John Maloney, uno de sus desarrolladores, pudiera ser ampliado para soportar proyectos mucho más grandes [4].

### Características del Diseño de Scratch

Los aspectos esenciales del diseño de Scratch se resumen en las siguientes características: *Metáfora de los bloques de construcción*: La programación con Scratch se basa en la metáfora de bloques de construcción, en la que los chicos construyen procedimientos encajando bloques gráficos como ladrillos de Lego o piezas de un puzzle.

*Manipulación de sonido, vídeo e imágenes*: La manipulación de archivos multimedia es una de las características más llamativas de Scratch, a las que se le han añadido rutinas de manipulación de imágenes en forma de filtros y control de los mismos.

*Profunda portabilidad*: Además de permitir compartir los proyectos a través de su plataforma en la web con un golpe de ratón, Scratch facilita el intercambio de gráficos (sprites) entre diferentes proyectos, autores e, incluso, entre diferentes tipos de dispositivos.

*Estrecha vinculación con el mundo físico*: El propósito de Scratch es programar objetos físicos del mismo modo que se programan objetos virtuales en pantalla usando entradas de sensores físicos para controlar los comportamientos de los objetos físicos y creaciones virtuales mediante el tablero (Scratch board) que están desarrollando en el MIT.

*Soporte para múltiples lenguajes*: Los

bloques de instrucciones de Scratch se pueden cambiar a diferentes idiomas sin ningún problema, incluso en plena ejecución, facilitando su uso en diferentes entornos lingüísticos.

### El Entorno de Desarrollo

El escritorio de Scratch se estructura en tres secciones principales, a izquierda, centro y derecha de la pantalla respectivamente: Paleta de bloques, Área de guiones (scripts) y Escenario. La programación se desarrolla en tiempo real y todas las modificaciones que realicemos en los guiones de los objetos se incorporan de forma automática en su ejecución. Como en Squeak, los modos de diseño y ejecución son simultáneos. No obstante, cuenta con un modo presentación que duplica las dimensiones del escenario. Esta función de visualización no está operativa, por el momento, en entorno Linux.

La paleta de bloques, situada en la zona izquierda de la pantalla alberga, categorizados por colores y funcionalidad, los bloques de programación arrastrables al área de scripts.

El menú principal ocupa la parte superior de la pantalla y permite realizar las operaciones comunes de crear, abrir y guardar proyectos, deshacer la última acción y proporcionar ayuda. Además, el botón *Extras* incluye la opción de establecer el lenguaje de los bloques de programación en diferentes idiomas.

Un botón interesante es el botón *Share* que nos permite, con un golpe de ratón, subir nuestros proyectos al espacio que previamente hayamos creado en la página web del proyecto. Una vez alojados en la web, los proyectos podrán ser ejecutados, mediante un player de java, sobre la misma página o descargados al ordenador local para

### Ejecutar Scratch en Linux

A pesar de lo publicado, Scratch sí se puede ejecutar en Linux y para ello nos bajaremos el paquete MS Windows desde [1] el paquete Scratch.exe. El problema con el que vamos a encontrarnos es que, al tratarse de un paquete exe, ni zip, ni tar, ni gunzip podrán abrirlo. La solución está en usar *Wine*. Si tenemos el emulador de Windows instalado, simplemente deberemos teclear en cualquier consola:

```
$ wine WinScratch
```

en el directorio donde hayamos descargado el programa. Esto ejecutará el proceso de instalación, creándose una carpeta *Scratch* que contiene todos los elementos del entorno.

Para poder acceder al entorno, simplemente tendremos que arrancar Squeak pasándole como parámetro la imagen que encontraremos en el directorio *Scratch*:

```
$ squeak Scratch.image
```

El resultado, si bien no es del todo satisfactorio, a falta de una versión oficial para Linux, nos permite hacernos una idea acerca de las prestaciones del programa. Una parte de sus funciones aún no están operativas bajo Linux, pero el uso general del programa obedece al usuario y nos permitirá familiarizarnos con el entorno a la espera de que sus desarrolladores realicen su lanzamiento definitivo para Linux.

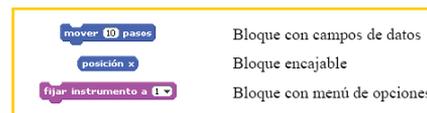


Figura 5: Tipos de bloques.

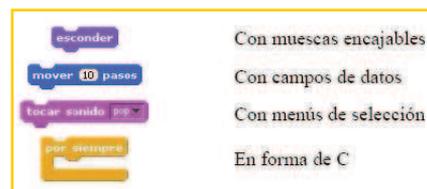


Figura 6: Bloques apilables.



Figura 7: Bloque cabecera.

abrirlos en la plataforma Scratch. De este modo se favorece el intercambio y el aprendizaje compartido de rutinas de programación, estudiando, modificando y mejorando otros programas.

### Sprites y Guiones

Los proyectos de Scratch están hechos de objetos llamados *sprites* que resultan fáciles de manipular. Los sprites son los gráficos con los que trabaja Scratch reconociendo los tipos de formatos más habituales (jpg, gif, bmp y png). Podemos cambiar el aspecto de un sprite dándole una forma o disfraz diferente, hacer que se parezca a una persona, un tren, una mariposa o cualquier cosa. Se puede usar cualquier imagen como disfraz, dibujar una imagen con el *Pintor*, importar una imagen de nuestro disco duro o, simplemente, arrastrarla desde una página web hasta el Escenario.

### Sonidos en Scratch

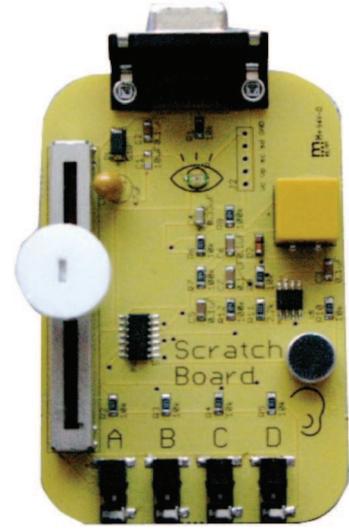
Los sprites pueden ejecutar sonidos seleccionados de la librería de sonidos pregrabados de Scratch, importarlos o realizar grabaciones mediante la grabadora de sonidos de Scratch. Una vez realizadas las grabaciones, los nuevos sonidos se incorporarán a la lista de los disponibles para el proyecto en el que estemos trabajando. Para ver los sonidos asociados a un determinado sprite haremos clic sobre la pestaña *Sounds*.

Scratch reconoce archivos MP3, archivos WAV, AIF y AU descomprimidos (codificados con 8-bits o 16-bits, pero no con 24-bits).

### Programar con Scratch

Lo que hace verdaderamente atractivo a Scratch es la gran simplicidad con la que, en muy poco tiempo, un usuario sin conocimientos de programación puede comenzar a elaborar y ejecutar sus propios proyectos.

El procedimiento consiste básicamente en incorporar sprites y fondos a la panta-



lla, bien arrojándolos sobre el escenario, bien creándolos mediante la aplicación *Paint Editor*, y dándoles vida arrastrando al área de *Scripts* bloques de programación.

La programación de los *sprites* se resume en proporcionarles instrucciones diciéndoles que se muevan, interpreten música o reaccionen a otros objetos. Para

# Descubre lo que te espera en la Red

Zona de descarga

Servicio al lector

Artículos descargables

Calendario de eventos



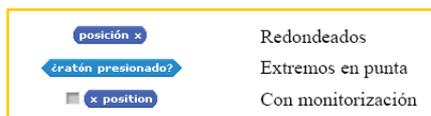


Figura 8: Informes.

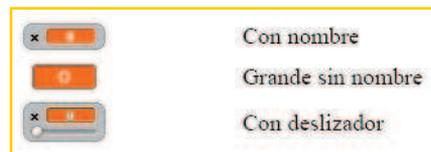


Figura 9: Tipos de monitores de variables.

indicarle a un *sprite* lo que tiene que hacer, debemos unir *bloques* gráficos formando pilas, denominadas *guiones*. Los guiones se construyen seleccionando, en primer lugar, un objeto o fondo de escenario en la lista de sprites y deslizando bloques desde la paleta de bloques y arrojándolos sobre el área de scripts. Crear guiones (programas) consiste, pues, en ir uniendo bloques gráficos formando una pila de instrucciones que se ejecutarán secuencialmente. Cuando se arrastra un bloque al Área de guiones, una marca blanca indica dónde se puede soltar el bloque y formar una conexión válida con otro bloque.

Para ejecutar un bloque y comprobar su funcionalidad bastará con hacer doble clic sobre cualquier parte de la pila, y se ejecutará todo desde el comienzo al final del guión. Además, existe la posibilidad de que Scratch nos muestre -en cada momento- el bloque que se está ejecutando, destacando cada bloque a medida que se ejecuta. Esta característica puede resultar muy útil para encontrar errores en los programas y para ayudar a los nuevos programadores a comprender el flujo de ejecución de un programa. Para habilitar la ejecución controlada de los guiones paso a paso seleccionaremos la opción *start single stepping* en el menú *Extras* del menú principal de Scratch.

### Tipos de Bloques

Algunos bloques tienen campos de texto editables, de forma redondeada, que permiten modificar sus valores escribiendo en su interior o arrojar sobre ellos bloques con esta misma forma. Otros tienen menús desplegables que facilitan la selec-



Figura 10: Ejecución de programas y guiones.

ción de opciones pulsando sobre la punta de flecha que oculta el menú.

En la paleta de bloques encontramos, clasificados en ocho categorías en función de su utilidad (movimiento, control, apariencia, sensores, sonidos, números, lápiz y variables), tres clases bloques:

**Bloques apilables:** Estos bloques tienen salientes en la parte inferior y huecos en la superior que permiten ajustar unos sobre otros formando pilas. Algunos de ellos tienen una zona de entrada en su interior, donde se puede escribir un número o elegir una opción de una ventana emergente. Un tipo especial de bloques tiene forma de C abierta en cuyo interior se pueden insertar otras pilas de bloques.

**Cabeceras:** Las cabeceras se distinguen por tener la parte superior redondeada, están destinadas a lanzar la ejecución de los guiones y se colocan en la parte superior de las pilas. Estos bloques esperan a que suceda un evento, como que se presione una tecla, se pulse la tecla intro o se haga clic sobre un *sprite*, para comenzar la ejecución de la pila de bloques que está debajo de ellos.

**Informes:** Estos bloques se han diseñado para que encajen en las zonas de entrada de otros bloques. Los informes se presentan con dos siluetas y encajan sólo en los huecos con la misma forma. Los informes con bordes redondeados representan números y encajan en bloques con huecos redondeados. Los informes con extremos terminados en punta informan valores booleanos (verdadero y falso) y encajan dentro de bloques con huecos con extremos en punta.

Los informes con monitorización en el escenario presentan un marcador junto a ellos. Al señalar el marcador se mostrará el contenido de la variable en el escenario. La forma de presentación de los informes monitorizados cambia al hacer doble clic sobre el propio marcador mostrando el nombre de la variable, con un marcador grande sin nombre o con un deslizador que permite manipular el valor de la variable. Este último formato (deslizador) únicamente está disponible para variables creadas por el usuario. Para ajustar los valores máximos y mínimos deberemos hacer clic derecho sobre el marcador de la variable con deslizador, seleccionar la opción *set sliden min and max* y establecer los valores sus valores extremos.

### Ejecución de Programas

La banderola verde situada sobre la esquina superior derecha del escenario permite comenzar la ejecución de todos los guiones

que comiencen con el encabezado *al presionar la bandera*. Alternativamente, mediante la tecla intro se obtiene el mismo resultado.

### Compartir con Scratch

Uno de los aspectos más interesantes de Scratch es la posibilidad de compartir ideas y proyectos con la comunidad de usuarios del programa, permitiéndonos estudiar los guiones de otros proyectos y modificarlos libremente. El botón *Share* presente en el menú principal del programa nos permitirá, mediante un golpe de tecla, subir nuestros proyectos a nuestro espacio personal en la página web del proyecto. Desde aquí podrán ser ejecutados on-line mediante un player de java o descargados para poder ser estudiados y modificados.

### Proyectos en Desarrollo

En la actualidad, el equipo Lifelong Kindergarten del MIT, desarrollador del proyecto, continúa mejorando el programa y está re-elaborando la página web de Scratch, que pretende ser una plataforma multilingüe que incluirá una biblioteca de *sprites* para que puedan ser utilizados libremente por sus usuarios.

Además de la plataforma Scratch que permita la compartición de proyectos, se está trabajando en un player para teléfonos móviles en J2ME que permita ejecutar los proyectos de Scratch en este soporte y en una versión completamente en línea del programa.

Por otra parte, se estudia la posibilidad de incluir en Scratch *shariables*, es decir, variables compartidas vía web.

Paralelamente, están trabajando en un tablero que incluye sensores de luz, sonido, deslizador y botones, además de cuatro resistencias de entradas para interruptores, sensores ligeros, de temperatura, etc. que permitirán actuar con el mundo exterior a través de Scratch. ■

**RECURSOS**

---

[1] Página web de Scratch: <http://scratch.mit.edu>

---

[2] Zona de descarga de Scratch: <http://llk.media.mit.edu/projects/scratch/download/>

---

[3] Scratch para informáticos en ciernes: <http://www.eecs.harvard.edu/~malan/publications/fp079-malan.pdf>

---

[4] Entrevista a John Maloney: <http://weeklysqueak.wordpress.com/2007/01/23/scratching-the-surface/>